

Radio Elettronica & Computer

7 programmi
su cassetta per
C64 e C128

Sped. in Abb. Postale Gr. III/70%

Anno XVII - Numero 8

Ottobre 1988 - L. 7.500

**GENERATORI
I PROGRAMMI
PER CREARE
VIDEOGAMES**

**ESPANSIONE
NON SOLO
GRAFICA**

**RIFLESSI
PRONTI!
L'hardware
per giocare
in tanti
al telequiz**

- 3 GIOCHI:**
- MISSIONE IMPOSSIBILE
 - DA SOLO NELLA GIUNGLA
 - SPIONAGGIO AEREO

**CARTUCCE
DUE ASSI
NELLA
MANICA**



Gruppo Editoriale
JCE

TASSA PAGATA PER CAMPIONE ALLEGATO

Pagina mancante (pubblicità)

Rubriche:

Software news pag. 6

Vorrei sapere,
vorrei proporre pag. 66

11 DUE ASSI NELLA MANICA

Hanno preso piede ormai ampiamente le cosiddette cartucce tuttofare, per le loro eccellenti prestazioni. In realtà, però, non tutti gli utenti ne hanno veramente bisogno. Per le esigenze dei più può essere sufficiente una cartuccia che...

15 I DODICI DELLA SQUADRIGLIA BLU

Ancora un gioco dedicato alla guerra del Vietnam. Questa volta, una squadriglia speciale va alla ricerca di un disertore fuggito in Cambogia con intenti spionistici...
Un inferno di fuoco.

17 OPERAZIONE DISCO

Quattro diversi caricatori all'opera. Vediamoli in azione, alle prese con i soliti, mai risolti problemi di duplicazione e trasferimento programmi da nastro a disco.

24 FLYING SHARK II

Nei cieli d'Africa, emulando le gesta del mitico Barone Rosso, una squadriglia di caccia affronta una missione difficile e divertente. Una serie di partite mozzafiato che metteranno alla prova i vostri nervi e la vostra abilità.

26 TRUCCHI E SCORCIATOIE

Cinque soluzioni, in questa puntata di "trucchi": una mini espansione per l'editor del C64; due utility che potenziano le prestazioni del DOS; un freezer per consentire un po' di riposo al computer; un'utility che controlla il drive.

28 ELVIN: MISSIONE IMPOSSIBILE

Nella sua insana perfidia, vuole distruggere il mondo. Ci riuscirà? Contro il perfido Elvin, in un dedalo di pericolose situazioni fuori dal tempo, a voi e solo a voi sta sconfiggere il folle criminale.

31 SHOOT'EM UP C.K.

Un programma eccezionale, che assieme a Game Maker vi consentirà di realizzare, a tu per tu con il C 64, fantastici videogames, grazie alle loro sorprendenti funzioni generatrici. E senza scrivere una sola riga di programma!

39 SG BASIC: NON SOLO GRAFICA

Notevole per la sua versatilità, Sprite & Graphic Basic è un'espansione che consente di facilitare al massimo la gestione degli sprite e dell'alta risoluzione, oltre a implementare nuove funzioni per la gestione dei programmi.

42 ALLA SCOPERTA DEI BUS

Continua la serie dedicata all'architettura dei bus; in questa puntata, uno sguardo ravvicinato all'attività del microprocessore: lettura ed esecuzione di tutte le istruzioni.

Sommario

49 INCREMENTO MINIMO

I trucchi della computazione lasciano sgomenti, a un primo approccio: calcoli che sembrano elementari vengono eseguiti dal computer secondo una logica procedurale che porta a risultati apparentemente assurdi, anzi, a veri e propri "errori". Eppure...

54 E CHE VINCA IL PIÙ...VELOX

L'appuntamento col fai da te di questo mese vi dà la possibilità di realizzare un simpaticissimo hardware che consente il colloquio contemporaneo, diretto e ordinato, di ben sette persone con il computer. Conoscete una situazione migliore per impostare un esilarante gioco a quiz? Oppure per...

62 MASTERCARD

Nel numero di luglio / agosto RE&C vi ha proposto un eccezionale lettore per gestire operazioni con tessere a strip codificato. Ebbene, con il software offerto in questo numero, potrete gestirlo al massimo delle sue possibilità!

Caricate così i programmi della cassetta allegata:

*Riavvolgete il nastro e digitate **LOAD** seguito da **RETURN** sulla tastiera del C64 e **PLAY** sul registratore. Verrà caricato il programma di presentazione con il menù dei programmi. Digitate **RUN** seguito dalla pressione del tasto **RETURN**. Terminata la presentazione, per caricare uno qualsiasi dei programmi è sufficiente digitare:*

LOAD "NOME PROGRAMMA"

*seguito dalla pressione del tasto **RETURN**.*

Responsabile Editoriale
Divisione Informatica
Francesca Marzotto

Direttore responsabile
Paolo Romani

Caporedattore
Marco Gussoni

Responsabile grafico
Desktop Publishing
Adelio Barcella

Impaginazione elettronica
Paola Bloise, Alessandra D'Ercole,
Mirko Diani, Giorgio Meroni

Collaboratori
Paolo Gussoni, Dolma Poli, Isa Sestini

Segretaria di redazione
Patrizia Angelo

Foto di copertina - Walter Battistessa

Testi, Programmi, Fotografie e Disegni
Riproduzione vietata Copyright.
Qualsiasi genere di materiale inviato in Redazione,
anche se non pubblicato
non verrà in nessun caso restituito.

RadioELETTRONICA&COMPUTER
Rivista mensile, una copia L. 7.500, numeri arretrati
il doppio del prezzo di copertina.
Pubblicazione mensile registrata presso il
Tribunale di Milano n. 112 del 17/3/72.

Fotolito: Bassoli - Milano.

Stampa: GEMM Grafica srl, Paderno Dugnano (MI).

Diffusione: Concessionario esclusivo per l'Italia
A.&G. Marco SpA, Via Fortezza 27 - 20126 Milano.
Spedizione in abb. post. gruppo III/70.

Abbonamenti: Annuale L. 60.000, estero L. 80.000.

RadioELETTRONICA & COMPUTER è titolare
in esclusiva per l'Italia dei testi e dei progetti di
Radio Plans e Electronique Pratique, periodici
del gruppo Société Parisienne d'Édition.

Gruppo Editoriale
JCE

Gruppo Editoriale JCE srl
Sede legale, Direzione, Redazione, Amministrazione
Via Ferri 6 - 20092 Cinisello Balsamo (MI)
Tel. 02/61.73.441 - 61.72.671 - 61.72.641 - 61.80.228
Telex 352376 JCE MIL I - Telefax 61.27.620

Direzione Amministrativa
Walter Buzzavo

Pubblicità e Marketing
Gruppo Editoriale JCE - Divisione Pubblicità
Via Ferri 6 - 20092 Cinisello Balsamo (MI)
Tel. 02/61.20.586 - 61.27.827 - 61.23.397 - 61.29.0038

**Concessionario esclusivo per Roma,
Lazio e centro sud:**
UNION MEDIA srl - Via G.B. Martini, 13
00198 Roma - Tel. 06/8119803-4
Telex 63026 UNION I - Telefax 06/5810537

Abbonamenti e Spedizioni
Daniela Radicchi

I versamenti vanno indirizzati a:
Gruppo Editoriale JCE srl, Via Ferri 6
20092 Cinisello Balsamo (MI), mediante l'emissione
di assegno circolare, cartolina vaglia o utilizzando il
c.c.p. n. 351205. Per i cambi di indirizzo allegare alla
comunicazione l'importo di L. 3.000, anche in francobolli,
e indicare insieme al nuovo anche il vecchio indirizzo.

Proprietario: *Editronica srl.*
Esercente l'impresa giornalistica:
GRUPPO EDITORIALE JCE srl.
Fanno parte del Gruppo Editoriale JCE srl:
JCE srl. - Editronica srl. - Edimedia srl.

Associato al

 **CSST** Consorzio
Stampa
Specializzata
Tecnica

Testata in corso di certificazione
obbligatoria secondo quanto stabilito
dal Regolamento del C.S.S.T.

 **USPI** Mensile associato
all'USPI
Unione Stampa
Periodica italiana

Geos

I sistemi operativi basati su icone stanno conoscendo un vistoso successo: gli utenti Amiga e Macintosh sanno bene quanto sia comodo lavorare con un mouse che controlla un puntatore per selezionare in modo immediato le icone presenti sullo schermo.

Anche gli affezionati del C64 possono facilmente adeguarsi a questo moderno modus operandi grazie a Geos della Berkeley Softworks.

Che cos'è Geos? È un vero e proprio sistema operativo alternativo a quello contenuto nelle ROM del C64, che fornisce al Commodore un aspetto decisamente Amiga-like, anche se con le doverose e inevitabili restrizioni.

Geos viene fornito su un floppy disk protetto, molto difficile ma non impossibile da copiare. È notevole comunque lo sforzo della Berkeley per evitare la proliferazione di copie pirata e ci risulta che qualche risultato sia stato raggiunto, infatti non è facile trovare copie non originali perfettamente funzionanti.

Passiamo ora a una descrizione dettagliata del pacchetto software. L'avvio del sistema operativo è molto semplice: basta inserire il disco originale nel drive e digitare LOAD"GEOS",8,1. È importante rilevare che il boot, ossia il caricamento iniziale del sistema, deve sempre essere effettuato dal disco originale, mentre qualunque altra operazione può essere effettuata con una copia del floppy. La Berkeley consiglia di effettuare subito una copia del disco prima di procedere

con qualunque altra operazione; infatti Geos fa un uso decisamente intenso del floppy disk, pertanto utilizzare l'originale può essere pericoloso poiché si rischia di perdere l'unico disco dal quale è possibile effettuare il boot.

Geos si presenta con una schermata in bianco e nero nella quale compare un riquadro dove si trova un certo numero di icone, ciascuna delle quali rappresenta un file sul disco, un menù a discesa, diviso in cinque sezioni, nonché alcune icone esterne al riquadro che rappresentano le periferiche connesse e gli eventuali file portati fuori dalla finestra.

Lavorare con Geos è davvero semplice e intuitivo: per caricare un file è sufficiente portare il puntatore sull'icona corrispondente servendosi del joystick e premere due volte il tasto Fire. Naturalmente non è detto che tutte le icone possano essere contenute in un solo riquadro, quindi ci sono diverse pagine-riquadro. Per voltare le pagine è sufficiente fare un clic sull'angolo della pagina.

Per selezionare un comando da menù si deve invece fare un clic su una delle sezioni, in modo che appaia l'elenco dei comandi, poi fare un clic sul comando desiderato. Come potete intuire è molto più difficile spiegare a parole come si compiono queste operazioni che imparare a eseguirle direttamente: l'uso di un sistema operativo a icone e menù a discesa è veramente semplice e pratico. Basta avere in mano il sistema per impadronirsi perfettamente della tecnica necessaria in breve tempo.

La schermata descritta che potete vedere in fotografia viene detta DeskTop. Da DeskTop è possibile effettuare qualunque operazione DOS, cioè è possibile gestire in modo completo i propri file e i propri floppy disk.

Vediamo quali sono i comandi disponibili dal menù:

- **Geos.** La prima sezione del menù è detta Geos, infatti riguarda alcuni comandi di preparazione del sistema: i primi due comandi servono per vedere il numero di versione di Geos e di DeskTop. Il comando Select Printer permette di selezionare la stampante connessa al computer. È disponibile una vastissima scelta di stampanti: praticamente è presente qualunque stampante adatta al C64. Mediante il comando Select Input è possibile selezionare il controllo del puntatore.



Il joystick di default è in porta 1, ma è possibile utilizzare anche il mouse, la tavoletta grafica e persino la penna ottica. L'ultimo comando di questa sezione serve per utilizzare eventuali utility presenti nel disco inserito nel drive.

- **File.** La seconda sezione si chiama File e permette le comuni operazioni di gestione file: il caricamento e l'esecuzione, attuabili anche con un doppio clic direttamente sull'icona, la copia, la ridenominazione e la visualizzazione dei dati relativi a un file.

- **View.** La sezione seguente è View e serve per selezionare la rappresentazione dei file nel riquadro di DeskTop. Normalmente ogni file è rappresentato da un'icona, tuttavia è possibile elencare i file semplicemente con il loro nome. In questo caso inoltre è possibile selezionare il criterio di ordinamento tra ordine alfabetico, ordine di data di registrazione, dimensione e tipo. Talvolta questa opzione risulta molto comoda, soprattutto quando si devono visualizzare directory molto lunghe.

- **Disk.** La quarta sezione si chiama Disk e permette la gestione dei floppy disk. È possibile per esempio aprire e chiudere un disco, dove per aprire si intende leggere la directory e mostrare nella finestra le icone relative ai file contenuti. Il disco di sistema è aperto automaticamente al boot, mentre se si desidera utilizzare un qualunque altro disco è necessario aprirlo. Viceversa quando si termina di utilizzare un disco è opportuno chiuderlo, ossia segnalare al sistema che non intendiamo più servircene. Da questa sezione è anche possibile cambiare il nome di un disco, effettuare il Validate, ossia controllarne l'integrità, nonché formattare un disco nuovo. Con l'istruzione Backup è possibile copiare un intero disco Geos, incluso il disco di sistema. Vi ricordiamo però che non è possibile effettuare il boot da una copia.

Se si dispone di un secondo drive, con numero di device 9, Geos lo riconosce durante il boot, ma se il vostro eventuale secondo drive ha numero di device 8 questo non è possibile. È sufficiente però servirsi dell'istruzione Add Drive per effettuare le opportune modifiche via software, in modo da disporre di un'ulteriore memoria di massa.

- **Basic, Reset, QLink.** L'ultima sezione del menù serve per tre comandi speciali: Basic, che consente di passare al Basic standard Commodore, Reset che consente la rilettura della directory di un disco in caso di errore, e QLink che permette il caricamento dell'omonimo programma che si trova sul lato B del disco Geos, e del quale parleremo in una prossima puntata della rubrica. Sul disco Geos, oltre al programma QLink che serve per la gestione di un modem, si trovano anche diverse utility. Le più importanti sono GeoWrite e GeoPaint, rispettivamente un word processor e un programma grafico.



Poiché si tratta di programmi molto sofisticati e complessi anche a essi dedicheremo maggiore spazio in un prossimo articolo. Altri accessori disponibili sono l'orologio-sveglia, la calcolatrice, NotePad, Photo Manager, Preference Manager e Text Manager. Consideriamo ora Geos più in generale: a che cosa serve? A nulla di preciso, perché in realtà non aggiunge niente di essenziale al C64 rispetto all'originale. Geos deve però essere considerato come un sistema operativo alternativo, profondamente diverso come concezione dalla impostazione rivolta essenzialmente al Basic del C64. Geos promuove il C64 a un livello superiore, è un'interfaccia utente molto più progredita dell'originale e fornisce al computer un sistema operativo

concettualmente simile ad Amiga. In realtà le similitudini tra Geos e Amiga workbench sono numerose e non si fermano all'impostazione generale.

La versione 1.3 di Geos funziona perfettamente anche su Amiga dotata di C64 Emulator e presenta la singolare caratteristica di essere l'unico turbo disk per 1541 funzionante su Amiga. È persino possibile utilizzare con successo il mouse di Amiga in emulazione del mouse 1351 del C64.

Naturalmente la velocità di elaborazione è vistosamente insufficiente, soprattutto perché Geos utilizza il modo alta risoluzione del C64 che richiede molto tempo per essere tradotto nella bitmap di Amiga. A questo proposito è necessario denunciare l'unico neo di Geos, che è proprio la lentezza di funzionamento.

Tutte le funzioni di DeskTop sono piuttosto lente, d'altra parte Geos sfrutta in modo alquanto sofisticato l'alta risoluzione del C64, e in effetti il 6510 (il microprocessore del C64) non è proprio un fulmine. È eccellente invece la gestione del floppy disk drive: è stato utilizzato un turbo molto veloce ed efficiente. Geos inoltre può funzionare anche con Speed-Dos inserito, anche se non si hanno variazioni di velocità. Geos è sicuramente un pacchetto software molto interessante, e nelle prossime puntate vedremo più in profondità come è possibile utilizzarlo, quando parleremo dei vari programmi scritti appositamente per Geos.

Giochi

Vediamo ora insieme qualche buon videogioco per l'incredibile e inesauribile C64.

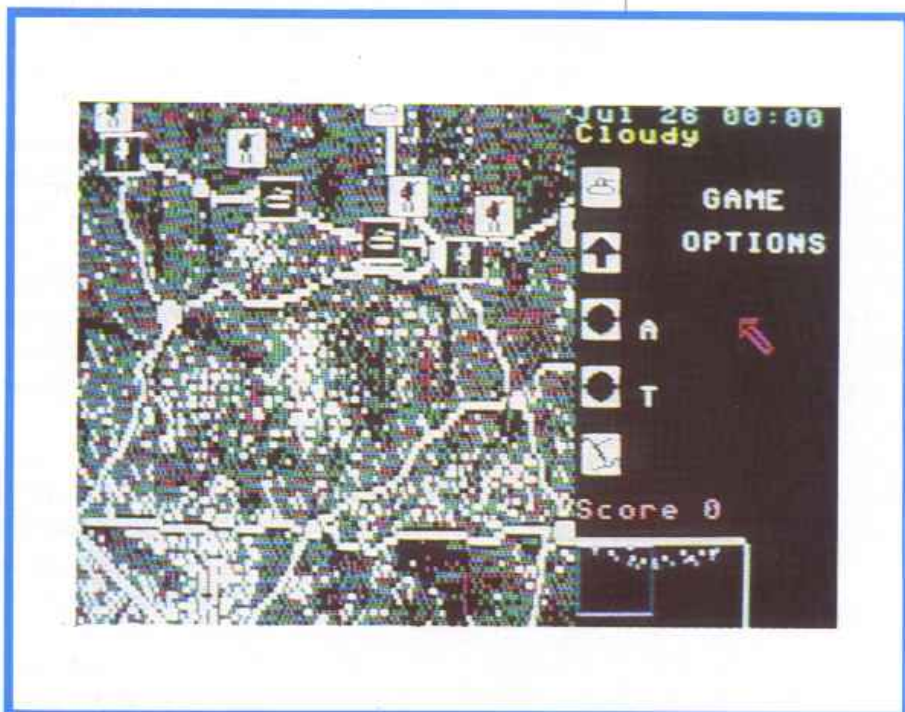
• **Patton vs Rommel.** Questo appassionante wargame della Electronic Arts prende spunto da uno dei momenti cruciali della seconda guerra mondiale: siamo infatti nel luglio del '44 e sul fronte francese le truppe alleate sbarcate in Normandia sferrano un massiccio attacco alle linee tedesche; l'offensiva, conosciuta come Operazione Cobra, mira al definitivo indebolimento dell'esercito germanico e successivamente alla liberazione dell'Europa intera dal giogo nazista. Vediamo la situazione dall'altro punto di vista: la supremazia della razza ariana sulle nazioni europee è insidiata dall'attacco in forze degli alleati che potrebbe compromettere il luminoso futuro del Terzo Reich. Qualunque sia il vostro punto di vista il programma in questione vi dà la possibilità di scegliere da che parte stare: se siete convinti che l'esito della guerra sia stato il più giusto opterete per le truppe alleate del generale Patton, se siete dei nostalgici della croce uncinata, o meglio, se amate il rischio, sceglierete di guidare le Panzer Divisionen assieme al maresciallo Rommel. Comunque vi schieriate, l'obiettivo consiste nell'assumere il controllo del maggior numero di città e di mantenerlo fino alla fine delle ostilità, cioè fino all'8 agosto (l'attacco incomincia il 25 luglio). Patton vs Rommel è studiato in modo da accontentare un pubblico con diverse esigenze: i tre livelli di gioco non si distinguono soltanto per la forza dell'avversario (comandato dal computer) ma soprattutto per le possibilità di comando disponibili, sempre più particolareggiate con il crescere del livello. In questo modo il principiante si limita a indirizzare le truppe che gli appaiono sulla mappa della Normandia, il giocatore più esperto può anche scegliere lo schieramento di battaglia e altri interventi strategici, infine al veterano è data la possibilità di intervenire direttamente sulla disposizione e sugli schieramenti in modo da risolvere le situazioni più intricate. Sulla cartina che rappresenta la zona delle operazioni appaiono i due eserciti contrapposti, formati dai relativi battaglioni; con un semplice spostamento del cursore di gioco sulla casella desiderata si può visualizzare ogni tipo di infor-



mazione necessaria a decidere se attaccare o no con una determinata truppa; si possono avere notizie sul tipo d'armata in questione (fanteria o artiglieria), sul numero di uomini di cui il battaglione dispone, sulle condizioni di una certa divisione e sul suo potenziale offensivo totale o disponibile al momento. Quando una truppa viene a trovarsi in una zona di dominio avversario (è possibile visualizzare anche i cerchi d'influenza) lo scontro è inevitabile: l'esito dipende dai fattori sopra citati ed eventualmente dall'appoggio aereo a uno dei due contendenti. Dilungarsi nei particolari di un wargame così completo può risultare soltanto dispersivo; per sottolineare la sua fattura basti pensare che a ogni divisione possono essere dati fino a 32 comandi, che verranno eseguiti contemporaneamente quando con l'apposita opzione si darà il via a una delle 15 giornate di battaglia. Alla fine di ogni giornata il computer fa un consuntivo delle operazioni militari verificatesi, analizzando le perdite e le vittorie su ambo i fronti e soprattutto i predomini sulle città; se il bilancio è negativo i tedeschi hanno avuto la meglio; viceversa se il punteggio è positivo le truppe alleate si sono dimostrate più abili; lo score sopra o sotto lo zero alla fine delle giornate di battaglia indicherà il vincitore. Patton vs Rommel è completato da alcuni aspetti che, pur mantenendo un carattere accessorio, risultano piacevoli per il giocatore; per esempio alla fine di ogni turno appare l'immagine di Patton o di Rommel (a seconda dell'esercito scelto) insieme ai consigli che i due insigni strateghi forniscono agli sprovveduti giocatori. Un'opzione molto utile a velocizzare il gioco permette di eliminare la visualizzazione dei combattimenti gli interventi dei due generali. La grafica non è eccezionale, ma in un gioco di questo tipo non è rilevante. Molto utile è invece la possibilità di salvare e ricaricare la partita in corso poiché la durata del gioco è abbastanza lunga.

• **The Games - Winter edition.** Ecco un gioco prodotto dalla Epyx: non abbiamo ancora finito di applaudire le fantastiche performances del nostro Alberto Tomba alle recenti Olimpiadi di Calgary ed ecco che sul mercato del software appare questa ennesima edizione dei giochi olimpici, estivi o invernali che siano. Probabilmente la Epyx ha puntato sull'onda-

ta di entusiasmo che queste gare hanno creato in milioni di spettatori, per riproporci in chiave abbastanza rinnovata quel Winter Games di non troppo remota uscita. Slittino, sci di fondo, pattinaggio artistico e di velocità, salto, slalom e discesa libera sono gli sport che questo gioco propone; trascurando la giocabilità (un paio d'ore sono più che sufficienti per archiviare il tutto), l'aspetto più apprezzabile di The Games è sicuramente la grafica, in particolare per alcune discipline come il pattinaggio artistico e il salto e per le cerimonie di apertura e chiusura ambientate nello stadio di Calgary. Come nelle altre edizioni si può competere a piacere in tutti o in alcuni sport, oppure



fare pratica provando e riprovando in una medesima disciplina; si può gareggiare contro otto avversari al massimo e scegliere il paese di appartenenza; è possibile infine visionare i record delle specialità e scegliere la porta del joystick. In conclusione The Games si potrebbe definire un gioco più da guardare che da giocare; del resto a chi si cimenta nelle varie discipline che esso propone non è richiesto altro che fare oscillare ritmicamente il joystick a destra e a sinistra (fondo, slalom e pattinaggio veloce) oppure capire qual è il momento giusto per premere il tasto Fire o abbassare la faticosa manopola (salto). L'unico elemento originale è offerto dal pattinaggio artistico, nel quale si può scegliere il tipo di brano di accompagnamento e il programma delle evoluzioni da compiere.

• **Imperium.** Un onesto arcade che ripropone l'ormai superato schema del salta l'ostacolo e fuggi è molto meno velleitario. Se volete diventare l'eroe che ha salvato il pianeta dalle forze nemiche (così si legge nella schermata di introduzione) dovrete saltare uno a uno gli strani esseri che compongono le agguerrite orde che vi si scagliano contro. In verità l'unico motivo che spinge il giocatore a cimentarsi con Imperium è l'estrema difficoltà con la quale si riesce a superare ogni gruppetto di alieni. Specialmente all'inizio bisogna provare e riprovare infinite volte per riuscire a far durare una partita più di dieci secondi. Il segreto sta tutto nell'individuare quegli automatismi che permettono, una volta scoperti, di procedere abbastanza agevolmente. Per esempio, non appena si è trovata l'esatta posizione in cui il protagonista deve stare per operare con successo un salto, e non appena si è trovato il giusto tempo per scavalcare gli ostacoli, è molto semplice aver ragione della prima schiera di alieni.

Per questo motivo la giocabilità risulta piuttosto limitata. La grafica è buona grazie anche alla simpatica immagine dell'astronauta rappresentata sullo schermo. La musica è discreta.

• **Thrust 2.** È un nuovo avvincente gioco prodotto dalla Ubik e dalla Firebird. Al comando di una navicella, vagate per i meandri del sottosuolo di uno strano pianeta dove gli alberi crescono anche all'ingù e dove dimorano le creature più svariate. In questo scenario il vostro com-

pito consiste nel recuperare uno a uno gli elementi che compongono una base spaziale per poi riportarli in superficie.

Questi componenti sono racchiusi in particolari sfere, i cui supporti sono simili a dei portauovo spaziali.

Non dovrete far altro che avvicinarvi, agganciare le sfere con un cavo d'acciaio ultrasensibile e depositarle sull'unica parte piana dell'insospitale luogo; istantaneamente esse si schiuderanno per originare una parte di edificio.

Il tutto sembrerebbe una sciocchezza se non intervenisse l'enorme forza di gravità che agisce sul pianeta, che rende problematico il comando della navicella.

Pilotare il mezzo diviene ancora più complicato quando a esso è collegata una sfera; le continue oscillazioni di quest'ultima rendono spesso inevitabile la collisione con le pareti rocciose circostanti. A ogni urto corrisponde una diminuzione del potere schermante degli scudi che proteggono la navicella; l'esaurimento della protezione comporta la fine della partita. Subito dopo l'aggancio di un componente avrete a disposizione un tempo limitato per riportare il carico in superficie.

Passando sulle caselle colorate sparse per lo scenario di gioco si ottiene la possibilità di sparare per un certo periodo.

La somma dei punti acquisiti, sia per i pezzi recuperati sia per le creature annientate, determina lo score finale.

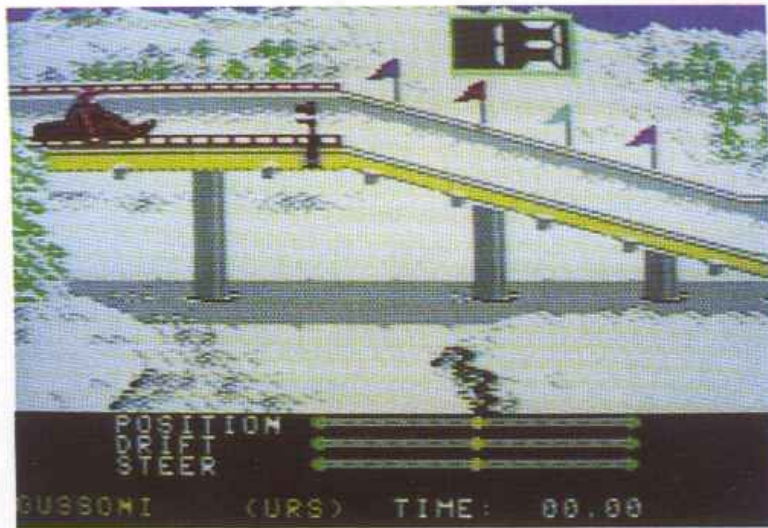
Questa telegrafica descrizione del gioco vi permette di apprezzare la discreta originalità del programma il quale, cosa che accade sempre più di rado, prende spunto da uno schema non ancora inflazionato dalla presenza di innumerevoli giochi di soggetto simile.

Se a questa qualità si aggiunge il realismo con il quale sono state riportate in un videogame le leggi della fisica (ci riferiamo ai rimbalzi della navicella e ai suoi movimenti dovuti all'oscillazione del carico), questo gioco merita veramente un buon voto.

La grafica e la musica sono adeguate all'insieme.

Unico neo è la mancanza di una mappa o di un radar per individuare la posizione in cui ci si trova e in cui si trovano le sfere: spesso capita di vagare a lungo per i meandri del pianeta senza trovare i tanto agognati componenti della base spaziale.

Gianni Arioli



Continua la serie delle recensioni di Radio Elettronica & Computer dedicate alle cartucce per Commodore 64. Se non vi serve una cartuccia tuttofare, tipo Final Cartridge, ma avete obiettivi ben precisi, quella giusta per voi probabilmente è...

Due assi nella manica

In questi ultimi mesi le cartucce per C64 hanno conosciuto un successo sempre crescente. Dopo i fast-load, che sono ormai un accessorio a cui nessun utente del C64 saprebbe rinunciare, sono comparse le cartucce freezer che permettono di copiare e modificare programmi aggirando un gran numero di protezioni.

Dopo le cartucce freezer hanno visto la luce le cartucce tuttofare, come la Final Cartridge III, della quale potete trovare la prova sul numero 5 di *Radio Elettronica & Computer*.

Questo tipo di cartuccia trasforma radicalmente il C64, mettendolo in grado di compiere operazioni impensabili.

D'altra parte bisogna considerare che talvolta disporre di una cartuccia sofisticatissima serve a poco: c'è un gran numero di sessantaquattresisti cui serve una cartuccia che sia semplicemente in grado di congelare ogni tipo di programma, permettendone eventuali modifiche.

A queste persone non serve una cartuccia multifunzione ma una cartuccia che faccia una cosa sola nel miglior modo possibile.

Questo mese quindi abbiamo provato ben due cartucce di quest'ultimo tipo: si tratta di Action Replay MK III e di The Expert.

Action Replay MK III

Action Replay consiste nella solita piccola scatola di plastica, da inserire nella porta cartridge del C64, più un esauriente manuale in inglese e un Enhancement disk, ossia un floppy disk con alcuni programmi accessori.

Sulla cartuccia si trovano i pulsanti di Reset e di Freeze. Il primo fornisce semplicemente il Reset hardware al C64, ossia connette il segnale di Reset del bus dati a massa, mentre il secondo permette di passare al menù principale che è composto di quattro opzioni.

La prima è proprio il Reset, e serve per configurare la memoria del computer per poter procedere nel modo migliore alle operazioni di congelamento. Alcuni programmi però riconoscono la memoria quando viene con-

figurata da una cartuccia, e in tal caso si bloccano. Per evitare questo inconveniente Action Replay consente di effettuare anche un altro tipo di Reset che lascia la RAM in uno stato random, rendendo meno efficienti le operazioni di compattazione del programma ma impedendo che un controllo sofisticato del software scopra la presenza della cartuccia e



quindi impedisca il lavoro.

Normalmente conviene sempre configurare la memoria prima di procedere al freezing di un programma, riservandosi la possibilità di ritentare con l'altro tipo di Reset qualora il primo tentativo fallisca.

Ripetere più volte il tentativo di congelamento non è un problema, infatti si tratta di un'operazione veloce e semplice: è sufficiente resettare il computer, caricare il programma da disco o da cassetta e premere nuovamente il tasto Freeze.

A questo punto ricompare il menù principale con due possibilità: premendo F3 il programma riprende a girare, come se non fosse stato premuto il tasto Freeze; premendo invece il tasto F5 viene presentato un secondo menù ricco di numerose possibilità.

Se desideriamo semplicemente effettuare il backup del programma basta selezionare l'apposita opzione e scegliere se il file deve essere inviato al floppy disk drive 8 o 9, oppure su nastro.

Se scegliamo il floppy il computer chiede se vogliamo registrare il file con un sistema turbo oppure alla velocità standard del 1541. Se invece vogliamo inviare su cassetta il file possiamo scegliere tra due possibili velocità di registrazione: la turbo, pari a circa cinque volte la velocità standard, oppure la superturbo, pari a circa il doppio della turbo.

A questo punto basta premere F1 per effettuare il Save sul dispositivo prescelto.

Se abbiamo scelto il floppy disk Action Replay crea un file di tipo PRG lungo fino a 245 blocchi che contiene in forma compattata l'intera memoria RAM del C64.

Se la lunghezza del file è inferiore a 202 blocchi è possibile caricarlo semplicemente con l'istruzione LOAD.

Se il programma che abbiamo congelato è piuttosto complesso, il file risultante sarà più lungo, e allora dovremo servirci per forza del caricatore che la cartuccia registra automaticamente sul disco

dove abbiamo effettuato il backup.

Questo caricatore inoltre permette il caricamento a velocità turbo, ossia circa cinque volte la velocità standard del 1541.

Il programma di compattazione di Action Replay non è molto potente, pertanto se disponiamo di un cruncher molto potente, come lo Speed Packer, è possibile provare a compattare ulteriormente il programma.

Da un file di 240 blocchi è facile ottenerne uno di soli 170-180 blocchi, con considerevole risparmio di spazio sul floppy disk, nonché di tempo di caricamento.

Finora abbiamo visto come si esegue un semplice backup, ma con Action Replay è possibile anche modificare un programma.

Per esempio è disponibile il cosiddetto Sprite Killer, che elimina l'effetto di collisione tra uno sprite e l'altro, oppure tra sprite e fondo.

Per servirsi dello Sprite Killer è sufficiente congelare il programma come se si dovesse effettuare un backup e selezionare l'apposita opzione dal menù.

Naturalmente sono possibili tre diversi tipi di Sprite Killer: è possibile eliminare le collisioni tra uno sprite e l'altro, tra sprite e fondo, oppure entrambe le cose.

L'effetto di questa operazione dipende dal gioco: con il glorioso Manic Miner per esempio si ottengono con facilità le cosiddette vite infinite, con altri giochi si ottiene solo confusione.

Un'opzione particolarmente interessante offerta da Action Replay consiste nella possibilità di cercare gli sprite all'interno del programma, in

modo da eliminarli o modificarli.

In realtà non c'è un vero e proprio Sprite Editor: Action Replay consente semplicemente di salvare i 64 byte che definiscono uno sprite su disco, nonché di ricaricarli in memoria.

Per modificarli è necessario servirsi di un'apposita utility, come la Pro-Sprite della Dosoft, oppure

procedere manualmente, definendo byte per byte lo sprite.

Gli amanti della Computer Graphic gioiranno per la presenza dell'opzione Picture Save, che consente di registrare schermate in modo multicolor in formato Blazing Paddles o Koala Painter.

Il primo formato può essere utilizzato anche per ottenere gli Slideshow, ossia per vedere successioni di disegni.

L'apposito programma è disponibile direttamente dal menù Utilities della cartuccia.

Action Replay dispone anche di un ridottissimo monitor che consente di effettuare pokes per eventuali modifiche dei programmi (tipo vite infinite, oppure inizio da uno schermo a scelta). È anche possibile visualizzare l'intera memoria del computer in codici esadecimali.

Non poteva mancare in Action Replay un mini fastload, che consente il caricamento da disco di qualunque programma a velocità cinque volte superiore allo standard.

Se si utilizza il fastload però si disabilita completamente il tasto Freeze, pertanto non è più possibile servirsi delle altre utility di Action Replay.

Possiamo dire che Action Replay è una cartuccia abbastanza potente e molto semplice da usare: la consigliamo senz'altro a chi desidera un freezer pratico ed efficace.

Ai più esigenti però consigliamo l'attenta lettura della prova di The Expert, che è sicuramente una cartuccia più sofisticata e complessa.

The Expert Trilogic

The Expert appare più sofisticata già a una prima occhiata infatti, oltre ai consueti comandi di Reset e di Freeze, troviamo un deviatore a tre posizioni nonché un led.

A questo proposito desideriamo informare i futuri acquirenti di questa cartuccia che si trova in commercio una versione maldestramente copiata; esigete sem-

pre la versione originale con il marchio Trilogic infatti alcune copie mancano del pulsante di Freeze e del led, pertanto non sono completamente funzionanti, inoltre sono dotate di un software che non funziona perfettamente.

The Expert si differenzia notevolmente da tutte le cartucce che abbiamo provato finora, infatti il software di gestione non è conservato in ROM bensì in RAM.

Naturalmente quando si inizia a usare la cartuccia è necessario programmare la RAM, per fare questo è sufficiente caricare da disco o da cassetta un apposito programma.

Qual è il vantaggio della RAM? È facile rispondere: poiché un floppy disk costa molto meno di una cartuccia, e poiché tutto il software della cartuccia è conservato su floppy, segue che per avere sempre l'ultima versione di cartuccia è sufficiente procurarsi il disco nuovo al costo di sole due sterline inglesi (meno di cinquemila lire).

È inoltre possibile programmare la cartuccia in modo diverso a seconda delle proprie esigenze; per esempio il software a noi fornito consiste di due diversi programmi per la cartuccia: uno da utilizzare per programmi singoli, l'altro per programmi multiloading che effettuano diversi caricamenti durante il funzionamento.

Ma vediamo come funziona e cosa permette di fare The Expert.

La prima operazione da compiere è la programmazione: per effettuarla basta inserire la cartuccia con il deviatore in posizione PRG, inserire il disco di sistema nel drive e caricare l'apposito programma: niente di più semplice.

Naturalmente è possibile effettuare le stesse operazioni anche da cassetta: d'ora in poi ogni volta che parleremo di disco sarà sottinteso che la stessa cosa vale anche per la cassetta.

Al termine del caricamento del software di sistema si deve spostare il deviatore in posizione On e premere il tasto Reset.

In questo modo si attiva il moni-

tor interno della cartuccia: per procedere con un freezing è sufficiente premere N e Return se si vuole configurare la memoria, oppure P e Return se è necessario effettuare un Reset random (valgono le stesse considerazioni effettuate per Action Replay).

In questo modo il computer si resetta e compare la classica schermata blu e azzurra.

A questo punto si porta il deviatore in posizione Off e si carica il programma voluto.

È importante notare che, finché il deviatore resta in posizione Off, la cartuccia praticamente non esiste o per meglio dire il computer non può accorgersi che la cartuccia è inserita, pertanto nessuna protezione anti-cartuccia può impedire il regolare caricamento e avvio del programma.

Per effettuare il Freeze è sufficiente riportare il deviatore su On e premere il tasto Restore, a meno che il led non si illumini, nel qual caso si deve premere il tasto di Freeze della cartuccia.

In realtà in tutti i programmi da noi provati non si è mai acceso il led, e tutte le operazioni si sono svolte regolarmente per mezzo del tasto Restore.

A questo punto ci si ritrova in modo monitor ed è possibile effettuare un vasto numero di operazioni; la qualità principale di questa cartuccia consiste infatti nel sofisticato monitor.

Digitando "Z"nome" è possibile salvare il programma in formato compactato, in modo da ottenere un file da caricare in seguito con una semplice istruzione Load, oppure con l'apposito loader, senza servirsi della cartuccia: si tratta del solito backup.

Se invece si desidera salvare o caricare una parte di memoria, è sufficiente servirsi delle istruzioni L e S.

Non mancano le consuete istruzioni per assemblare brevi routine in linguaggio macchina, o per disassemblare il programma in memoria; inoltre il monitor fornisce l'indirizzo dell'istruzione in



esecuzione al momento del Freeze, pertanto è possibile controllare che cosa stava eseguendo il computer prima che la cartuccia fosse azionata.

C'è anche un'ottima istruzione Hunt che permette di trovare sequenze di byte espresse in codici esadecimali, in codici di schermo oppure ASCII.

L'istruzione più importante però consiste nella possibilità di fare ripartire il programma con grande facilità esattamente dal punto dove era stato fermato.

Questa possibilità è disponibile anche con Action Replay, tuttavia in quella cartuccia non è possibile servirsi con facilità a causa della mancanza di un monitor efficiente. Non è nostra intenzione dilungarci ulteriormente elencando le istruzioni disponibili, tuttavia possiamo affermare con certezza che questo monitor è veramente molto potente e efficiente, quindi permette di modificare il programma nel modo migliore.

Il giudizio complessivo su The Expert è decisamente positivo: questa cartuccia è senz'altro adatta all'utenza più esperta, poiché richiede una certa dose di esperienza a differenza di Action Replay che è molto intuitiva.

In cambio però fornisce la possibilità di congelare un grandissimo numero di programmi, inoltre permette di modificare un programma con facilità grazie al potentissimo monitor e alla possibilità di riattivare il programma dopo ogni modifica.

Gianni Arioli

Pagina mancante (pubblicità)

Nulla a che vedere con una calma e tranquilla foresta equatoriale: un vero inferno di fuoco metterà a dura prova la vostra abilità.

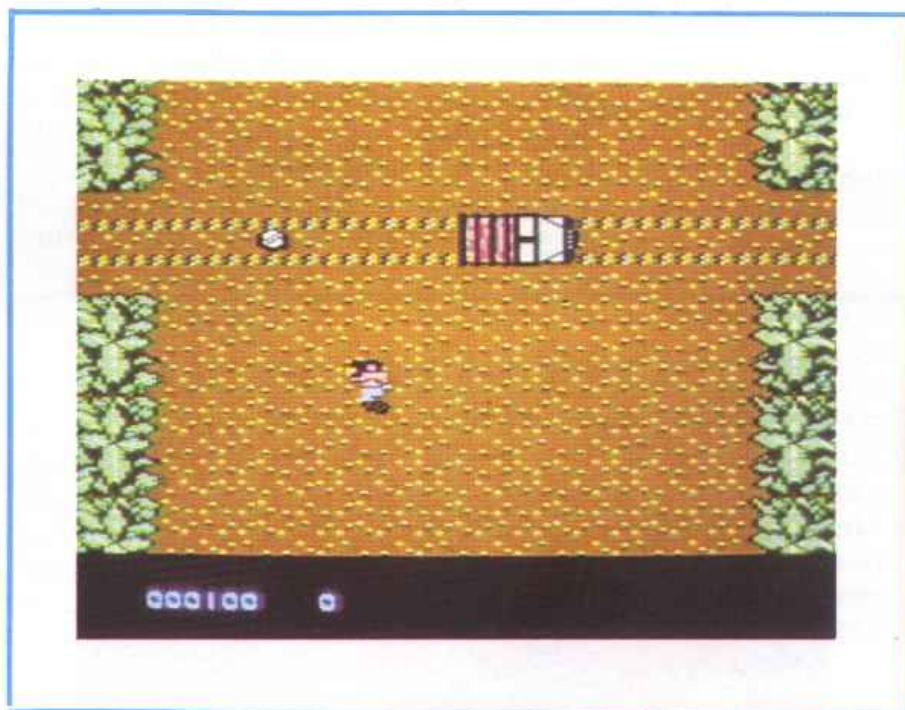
I dodici della squadriglia blu

Era da parecchio tempo che non si vedeva un gioco del genere: questo infatti riprende la guerra del Vietnam. Questa guerra civile, che durò quasi vent'anni, ebbe inizio nel 1956 nel Vietnam del sud per le mancate elezioni politiche che dovevano portare all'unificazione con il Vietnam del nord. Come in ogni guerra c'era chi vinceva, chi soccombeva, e... chi ci speculava. Intorno agli anni sessanta, si sviluppò infatti il mercato nero: uno dei boss di questo commercio clandestino era il tenente colonnello Marok, un disertore dell'esercito americano che si era rifugiato nella catena montuosa dell'Annam vicino alla costa bagnata dal mar cinese meridionale. Pochi anni più tardi, si venne a sapere che Marok era riuscito a fotografare in una base USA dei piani segreti riguardanti il progetto di una base missilistica. Disertando e rifugiandosi in Cambogia, portò le fotografie con sé e lì cercò di venderle al migliore offerente organizzando un'asta. Scoperto il suo gioco, gli americani corsero ai ripari e inviarono sul posto una squadra speciale, attrezzata e allenata per l'operazione: la Squadriglia Blu. Era formata da dodici uomini addestrati a far fronte a tutte le insidie, capaci di combattere su ogni terreno, sia in mare sia in montagna... La loro missione era di trovare Marok e le sue fotografie e di riportarlo in patria. Arrivati sul bersaglio con un bombardiere, si lanciarono col paracadute. Sfortunatamente per loro, furono avvistati dagli uomini di Marok che organizzarono una trappola. Questa riuscì perfettamente ed essi furono tutti catturati tranne il capo della Squadriglia Blu, che sfuggì mimetizzandosi nella jungla come solo lui sapeva fare. I prigionieri furono portati nella base

del traditore e lì legati a dei pali. Nel frattempo il capo squadriglia decise di lasciar passare la notte e di attaccare l'accampamento il mattino dopo. Ce la farà a salvare i suoi amici, a recuperare i piani, a catturare il traditore e a ritrovarsi all'ora prestabilita con l'elicottero americano che lo doveva riportare in patria?

Come si gioca

Il giocatore è armato di un potentissimo mitra a carica praticamente illimita-



ta. Lo scrolling è di tipo verticale e la vista è dall'alto. Nell'avanzare egli si dovrà inevitabilmente scontrare contro uomini armati che arrivano da tutte le parti, uomini dietro le trincee, ordigni pericolosissimi che sparano all'impazzata... il tutto in un impervia foresta equatoriale. Suc-

cessivamente, il capo squadriglia si troverà su un ponteggio sopra la palude e, poco dopo libererà i primi quattro ostaggi. Sul percorso, in alcuni tratti, sono presenti dei bivi che determineranno la sorte dei prigionieri; se infatti il capo deciderà di prendere una strada piuttosto che un'altra, essi saranno abbandonati al loro destino nelle mani del traditore. In



punti strategici arriveranno velocemente elicotteri dai lati dello schermo. Questi sofisticati mezzi da combattimento lasceranno cadere sopra la testa dell'omino grappoli di bombe, che esplodendo a terra genereranno acuminate schegge. Se queste colpiranno l'omino gli faranno



perdere una delle tre vite a disposizione.

Per arrivare alla fine della missione si devono superare quattro livelli successivi di crescente difficoltà. Al termine di ognuno si trovano due portoni, nei quali passa corrente elettrica, per superare i quali si deve sparare fino a distruggerli.

Il paesaggio si muove con l'omino, cioè se proseguite nel cammino esso scrollerà dall'alto verso il basso. In alcuni tratti però si fermerà rimanendo fisso per alcuni secondi e la vostra abilità sarà messa a dura prova. Il finale è abbastanza buono ma forse non troppo appagante dopo tutte le fatiche fatte per arrivarci. La schermata iniziale presenta una discreta grafica e il gioco nel complesso è divertente.

Purtroppo però non ha opzioni, può giocare una sola persona alla volta e non c'è la possibilità di predisporre vite infinite.

Non c'è sottofondo musicale ma gli effetti sonori suppliscono bene a questa mancanza. La difficoltà è abbastanza elevata e non aumenta linearmente, infatti durante il percorso si alternano punti facili ad altri praticamente insuperabili.

La grafica è discreta, anche gli sprites sono ben disegnati e ve ne è una buona varietà, simpatico è l'effetto dissolvenza.

Consigli e conclusioni

Vi consigliamo di procedere molto lentamente e di arrestarvi se vedete che troppi nemici appaiono sullo schermo. Ricordatevi che il vostro scopo è di arrivare in fondo e non di distruggere tutto ciò che si muove; perciò è meglio aver salva la vita cercando di evitare i colpi piuttosto che rischiarla insistendo nel distruggere avamposti. Al primo bivio ricordatevi di svoltare a sinistra. Le orde di nemici si presentano sempre con la stessa cadenza e provengono dalle stesse direzioni in tutte le partite: cercate quindi di memorizzare questa sequenza. Colpiti dai vostri spari, alcuni nemici scompariranno al primo colpo, mentre altri dovranno essere colpiti numerose volte. Le basi sono disposte in maniera tale che, per colpirne una, dovete entrare nella zona di fuoco di un'altra, perciò state molto attenti. I nemici a piedi sparano nella direzione in cui camminano, cioè in orizzontale o in verticale o solo in diagonale; pochissimi sparano dove capita.

Per iniziare inserite il joystick nella porta 2 e date il via alla partita.

Oscar Maeran

In questo nuovo appuntamento sull'argomento duplicazione e trasferimento dei programmi da nastro a disco vedremo in azione quattro caricatori.

Operazione disco

La caratteristica comune a questi caricatori è che, al contrario del caricatore visto nella scorsa puntata (caricatore 0), in questi quattro le routine di Load Turbo sono posizionate in quella zona di memoria, chiamata buffer del registratore, che viene riempita con i dati dell'header del programma su nastro.

La possibilità di sfruttare aree di memoria, che non entrano in conflitto con quelle usate dal programma caricato e inutilizzate dal S.O., permette di caricare programmi con estensioni notevoli, tanto per intenderci di 202 blocchi sul disco, dalla
locazione
\$0801 alla
\$CFFF,

mette, come nel caricatore 4, di riabilitare il video per visualizzare schermate grafiche o messaggi.

Le diversità tra un caricatore e un altro ci impediscono di creare un metodo unico di lavoro; di conseguenza nasce la necessità di listare e disassemblare la memoria relativa ad ognuno dei caricatori che si vuole sproteggere (vedi i listati seguenti) e quindi, in ogni caso, dobbiamo eseguire le operazioni viste col caricatore 0 e che comunque riepiloghiamo:

a) Innanzitutto occorre un programma monitor, la volta precedente abbiamo usato lo Zoom, che viene attivato con SYS 49152.

b) Si può pulire la memoria con il co-

mando F 0800 BFFF 00, in modo da avere un'idea immediata del tipo di caricatore che si ha in memoria e della sua estensione.

c) Effettuare il Load del programma dal nastro con il comando L "" 01 1000.

d) Una volta fermo il registratore, lasciando il tasto Play premuto, possiamo analizzare con lo Zoom l'eventuale programma in memoria, decidendo come agire.

Primo caricatore

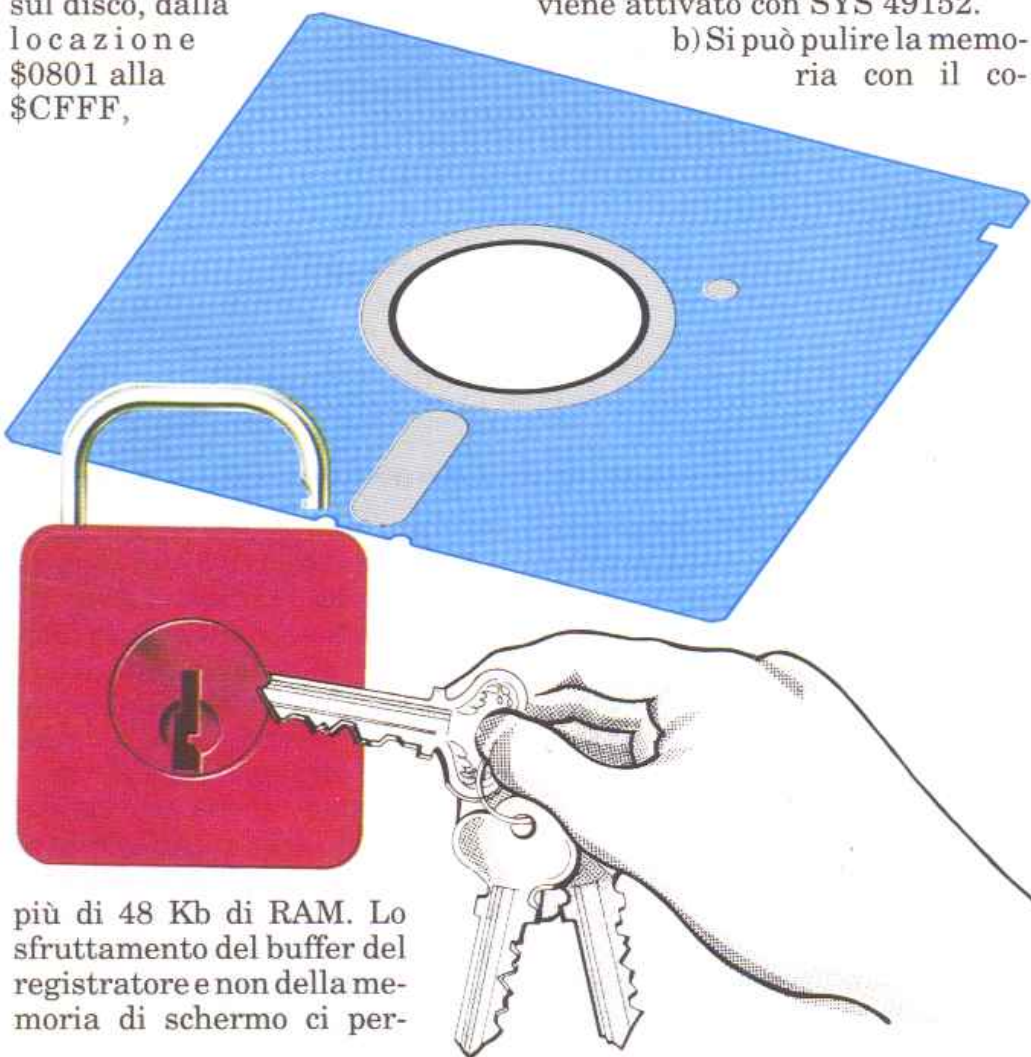
Il **listato 1** mostra un caricatore che si avvicina molto, come struttura generale, a quello 0, infatti, anche se è diviso in due parti, possiamo distinguere:

1) il loader turbo con tutte le routine necessarie, nel buffer del registratore;

2) la parte che si occupa della preparazione dei puntatori e dell'attivazione del programma una volta caricato (che è quella che noi troviamo dall'indirizzo \$1000, ma che in origine viene posta agli indirizzi specificati nel buffer del registratore, \$033D/\$033E per l'inizio e \$033F/\$0340 per la fine).

L'indirizzo di inizio di questa seconda parte, che normalmente è \$02BA, corrisponde ancora a una parte della memoria del C64 inutilizzata dal S.O., cioè da \$02A7 a \$02FF.

Osservando l'estensione del programma caricato dalla locazione \$1000 notiamo che eccede lo spazio a disposizione da \$02BA a \$02FF; questo perché è presente con esso la tabella dei vettori di sistema, modificata opportunamente, la quale viene sovrapposta, al momento del Load, a quel-



più di 48 Kb di RAM. Lo sfruttamento del buffer del registratore e non della memoria di schermo ci per-

la esistente in modo da ottenere l'esecuzione automatica del caricatore. Se infatti controlliamo i codici e gli indirizzi elencati da \$1046 in avanti e li confrontiamo con quelli riportati sul manuale del C64 (pag.161-166), dall'indirizzo \$0300 in avanti, possiamo notare delle corrispondenze e delle differenze tra le due tabelle, cosa che ci fa pensare al sistema di Autostart visto con il caricatore 0.

Basterebbe quindi modificare il listato in questione cambiando il RUN (R [shift] U) con un END (E [shift] N), come con l'altro caricatore, e trasferire il programma da \$1000 a \$1045 nella memoria originale a \$02BA. Vi accorgete però che il caricatore non funziona, o meglio, funziona ma non carica tutto il programma. La causa di ciò è il mancato settaggio di un vettore; infatti alla locazione \$1009 e alla \$100E c'è il caricamento di due celle di memoria, precisamente la \$62 e la \$63. Esse vengono settate prelevando il dato da locazioni (dalla \$030C e \$0313 che corrispondono alla \$1052 e \$1059 rispettivamente) che sono oltre quelle trasferite; noi infatti spostiamo un pezzo di memoria da \$1000 a \$1045 mettendolo da \$02BA a \$02FF, tralasciando quei due bytes tanto importanti:

Vi accorgete però che trasferire il blocco da \$1000 a \$1059, comprendendo i bytes che ci interessano, vi porterà ad un blocco del sistema. Questo problema è facilmente risolvibile con una semplice modifica del listato che ora eseguiamo insieme:

- Visualizzate la memoria da \$1046 a \$1060 e annotate i codici presenti alle locazioni \$1052 e \$1059.

- Disassemblate quindi da \$1000 a \$1015 e posizionatevi sul codice esadecimale dell'istruzione LDA \$030C all'indirizzo \$1009, cioè AD, cambiandolo in A9; modificate anche il byte successivo, 0C, scrivendo al suo posto il dato letto all'indirizzo \$1052 che nel nostro listato è \$90, al termine premete Return.

Listato 1. Caricatore 1.

```

033C 03 BA 02 2C 03 4C 4F 47 `.....LOG
0344 49 43 41 20 34 30 41 20 `ICA 40A

0361 ad 11 d0 lda $d011      disabilita
0364 29 ef and #$ef         il
0366 8d 11 d0 sta $d011      video
0369 a0 00 ldy #$00
036b 84 c0 sty $c0          fa partire il motore del registratore
036d ca dex
036e d0 fd bne $036d      azzerramento registri con ciclo di ritardo
0370 88 dey
0371 d0 fa bne $036d
0373 78 sei
0374 84 d7 sty $d7
0376 a9 07 lda #$07        setta registro low TIMER B
0378 8d 06 dd sta $dd06
037b a2 01 ldx #$01
037d 20 e5 03 jsr $03e5
0380 26 bd rol $bd
0382 a5 bd lda $bd
0384 c9 02 cmp #$02
0386 d0 f5 bne $037d
0388 a0 10 ldy #$10
038a 20 d2 03 jsr $03d2      tra
038d c9 02 cmp #$02
038f f0 f9 beq $038a
0391 c4 bd cpy $bd         COMPUTER e TAPE
0393 d0 e8 bne $037d
0395 20 d2 03 jsr $03d2
0398 88 dey
0399 d0 f6 bne $0391
039b 20 d2 03 jsr $03d2      riceve byte da nastro
039e 91 26 sta ($26),y      memorizza in ram
03a0 d1 26 cmp ($26),y      se c'e' un errore
03a2 f0 02 beq $03a6      attiva load error
03a4 86 90 stx $90
03a6 45 d7 eor $d7
03a8 85 d7 sta $d7
03aa e6 26 inc $26
03ac d0 02 bne $03b0
03ae e6 27 inc $27
03b0 a5 26 lda $26         incremento puntatori di
03b2 c5 62 cmp $62
03b4 a5 27 lda $27         memorizzazione programma
03b6 e5 63 sbc $63
03b8 90 e1 bcc $039b
03ba 20 d2 03 jsr $03d2
03bd a0 00 ldy #$00
03bf ca dex
03c0 d0 fd bne $03bf      azzera registri x e y
03c2 88 dey
03c3 d0 fa bne $03bf      con ciclo di ritardo
03c5 c8 iny
03c6 84 c0 sty $c0        ferma il registratore
03c8 58 cli
03c9 18 clc
03ca a9 00 lda #$00
03cc 8d a0 02 sta $02a0
03cf 4c 93 fc jmp $fc93   ritorno con ripristino vettori
03d2 a9 08 lda #$08
03d4 85 a3 sta $a3       ROUTINE DI RICEZIONE BYTE
                                memorizza vettore di conteggio bit

```

Noterete che l'istruzione cambierà da LDA \$030C a LDA #\$90, cioè da un caricamento dell'accumulatore di tipo diretto ad uno di tipo immediato.

Avanza un byte da questa operazione di modifica, che contiene 03; infatti le due istruzioni in memoria sono così codificate:

AD 0C 03 per LDA \$030C
A9 90 per LDA #\$90

per cui dobbiamo eliminare quel byte in più, inserendo per esempio

un'istruzione NOP, codice \$EA, scrivendolo sulla riga alla quale vi trovate dopo l'ultimo Return. È consigliabile però cancellare tutti i caratteri alla destra del codice appena scritto, prima di premere di nuovo Return, onde evitare spiacevoli sorprese. Ripetete queste operazioni anche con l'istruzione LDA \$0313 all'indirizzo \$100E, inserendo il dato della locazione \$1059. Al termine, disassemblate di nuovo la memoria da \$1000 a \$1015 e vi ritroverete con il listato così modificato:


```

03d6 20 e5 03 jsr $03e5      riceve bit di ritorno nel carry
03d9 26 bd    rol $bd      che viene messo in $BD con uno shift
03db ea      nop
03dc ea      nop
03dd ea      nop
03de c6 a3    dec $a3      decrementa contatore bit
03e0 d0 f4    bne $03d6     ripete fino a 0
03e2 a5 bd    lda $bd      carica accumulatore con il byte ricevuto
03e4 60      rts
03e5 a9 10    lda #$10      ROUTINE DI RICEZIONI DI UN BIT DA REGISTRATORE
03e7 2c 0d dc bit $dc0d     attende il fronte dal tape
03ea f0 fb    beq $03e7
03ec ad 0d dd lda $dd0d     legge registro di stato del 6526
03ef 8e 07 dd stx $dd07     memorizza nuovamente registro high timer B
03f2 48      pha          salva accumulatore in stack
03f3 a9 19    lda #$19
03f5 8d 0f dd sta $dd0f     fa ripartire il timer
03f8 68      pla          ricarica dato da stack
03f9 4a      lsr          sposta flag timer B
03fa 4a      lsr          in carry
03fb 60      rts

1000 00 09 73 a4
1004 a5 2b    lda $2b      setta puntatori di inizio save
1006 ea      nop
1007 85 26    sta $26      con puntatore inizio area basic ($0801)
1009 ad 0c 03 lda $030c     carica puntatore high fine load
100c 85 63    sta $63      salva in vettore high
100e ad 13 03 lda $0313     carica puntatore low fine load
1011 85 62    sta $62      salva in vettore low
1013 a5 2c    lda $2c      puntatore inizio load high
1015 ea      nop
1016 85 27    sta $27
1018 20 61 03 jsr $0361     LOAD TURBO
101b 20 15 fd jsr $fd15     inizializza vettori di I/O
101e 20 53 e4 jsr $e453     inizializza vettori
1021 20 33 a5 jsr $a533     relink basic
1024 a5 62    lda $62      setta puntatore di
1026 a6 63    ldx $63      inizio delle variabili
1028 85 2d    sta $2d      con puntatore di fine
102a 86 2e    stx $2e      programma
102c a9 52    lda #$52     =====routine di autorun=====
102e 8d 77 02 sta $0277     salva r in buffer tastiera
1031 a9 d5    lda #$d5     carattere shift u
1033 8d 78 02 sta $0278     salva shift u in buffer tastiera
1036 a9 0d    lda $0d      return
1038 8d 79 02 sta $0279     salva return in buffer tastiera
103b a9 03    lda $03      inserisce 3 in contatore caratteri presenti
103d 85 c6    sta $c6     nel buffer della tastiera
103f 4c 74 a4 jmp $a474     ritorno al S.O.
1042 ea      nop
1043 ea      nop
1044 ea      nop
1045 ea      nop
1046 be 02 be 02 7c 01 14 a7
104e e4 a7 86 ae 90 62 00 32
1056 4c 48 b2 01 2c f9 57 08
105e e2 fc 4a f3 91 f2 0e f2
1066 50 f2 33 f3 fb 03 fb 03
106e b1 f1 fb

```

```

1000 00      BRK
1001 09 73   ORA #$73
1003 A4 A5   LDA $A5
1005 2B      ???
1006 EA      NOP
1007 85 26   STA $26
1009 A9 90   LDA #$90 <=====
100B EA      NOP
100C 85 63   STA $63
100E A9 01   LDA #$01 <=====
1010 EA      NOP
1011 85 62   STA $62

```

L'importanza di questi due bytes sta nel fatto che essi determinano la fine del Load e quindi del programma in caricamento.

Perciò un'alterazione o un mancato settaggio del vettore \$62/\$63

porta a un arresto del caricamento del programma prima della fine effettiva. Dopo aver fatto questa modifica cambiate il RUN, che attiva il programma a fine Load, con un END; operazione che potete effettuare cambiando i codici:

52 (codice ASCII della lettera R) alla locazione \$102D con 45 (codice ASCII di E)

D5 (codice ASCII di U [shift u]) alla locazione \$1032 con CE (codice ASCII di N [shift n]).

Finalmente potete trasferire il programma correttamente modificato usando il comando:

T 1000 1045 02BA

attivandolo poi con un G02BE.

Un vantaggio che si ha rispetto al caricatore 0 è che è possibile conoscere, prima di caricare il programma, il suo indirizzo di fine, potendo così predeterminare quale metodo usare per salvare il programma su disco.

In ogni caso è consigliabile usare la routine di Dsave, migliorata in alcuni punti rispetto la precedente versione, che permette un salvataggio del programma su disco in un modo più sicuro e aiuta anche, come vedremo poi, nella protezione di un programma.

Questa routine si colloca nella memoria video, si presenta come un insieme di strani caratteri che appaiono sullo schermo e utilizza, come nome da assegnare al file da salvare, i sedici caratteri in corrispondenza della scritta Nome del File. L'attivazione avviene scrivendo, dopo la chiamata alle routine di Load, indirizzo \$1018, l'istruzione di salto all'indirizzo \$0410, JMP \$0410 (4C 10 04 i codici operativi).

Eseguita questa operazione, unitamente alle modifiche delle istruzioni LDA \$030C e LDA \$0313 viste prima, possiamo trasferire la memoria da \$1000 a \$1045, successivamente eseguiamo il caricamento della routine Dsave, avendo cura di digitare il comando di Load a metà schermo circa per evitare così eventuali cancellazioni.

Poi scriviamo il nome del file e infine attiviamo il caricatore con G02BE, facendo attenzione a non causare lo scroll del video.

Se volete un consiglio, createvi un disco di lavoro trasferendovi i programmi di Dsave, lo Zoom e le routine più importanti.

Secondo caricatore

Confrontiamo ora il caricatore 2 con quello appena visto eviden-

Listato 2. Caricatore 2

033c 03 ba 02 2c 03 56 49 41'....,via			
0344 47 47 49 4f 20 4e 45 4c'ggio nel			
034c 20 54 45 4d 50 42 49 54' tempbit			
0354 55 52 42 4f 20 42 59 20'urbo by			
035c 53 43 20 38 35 ad 11 d0'sc 85...			
0361 ad 11 d0 lda \$d011	disabilita schermo		
0364 29 ef and #\$ef			
0366 8d 11 d0 sta \$d011			
0369 a0 00 ldy #\$00	attiva motore		
036b 84 c0 sty \$c0	registratore		
036d ca dex			
036e d0 fd bne \$036d			
0370 88 dey			
0371 d0 fa bne \$036d			
0373 78 sei			
0374 84 d7 sty \$d7			
0376 a9 07 lda #\$07	setta registro low timer B		
0378 8d 06 dd sta \$dd06			
037b a2 01 ldx #\$01			
037d 20 e5 03 jsr \$03e5	riceve bit da nastro		
0380 26 bd rol \$bd	sposta bit in \$bd		
0382 a5 bd lda \$bd			
0384 c9 02 cmp #\$02			
0386 d0 f5 bne \$037d	verifica carattere di sincronismo		
0388 a0 10 ldy #\$10	sincronismo di inizio programma		
038a 20 d2 03 jsr \$03d2	riceve una sequenza di bytes per sincronizzarsi		
038d c9 02 cmp #\$02	con l'inizio del programma e sono 16 byte contenenti		
038f f0 f9 beq \$038a	codici decrescenti da 10 a 0		
0391 c4 bd cpy \$bd			
0393 d0 e8 bne \$037d			
0395 20 d2 03 jsr \$03d2			
0398 88 dey			
0399 d0 f6 bne \$0391			
039b 20 d2 03 jsr \$03d2	riceve byte programma		
039e 91 26 sta (\$26),y	memorizza in ram		
03a0 d1 26 cmp (\$26),y	verifica se c'e' un errore		
03a2 f0 02 beq \$03a6			
03a4 86 90 stx \$90	setta codice per load error		
03a6 45 d7 eor \$d7			
03a8 85 d7 sta \$d7			
03aa e6 26 inc \$26			
03ac d0 02 bne \$03b0			
03ae e6 27 inc \$27			
03b0 a5 26 lda \$26	incremento puntatori		
03b2 c5 62 cmp \$62			
03b4 a5 27 lda \$27			
03b6 e5 63 sbc \$63			
03b8 90 e1 bcc \$039b			
03ba 20 d2 03 jsr \$03d2			
03bd a0 00 ldy #\$00			
03bf ca dex			
03c0 d0 fd bne \$03bf			
03c2 88 dey azzera x e y con ciclo di ritardo			
03c3 d0 fa bne \$03bf			
03c5 c8 iny			
03c6 84 c0 sty \$c0	ferma registratore		
03c8 58 cli			
03c9 18 clc			
03ca a9 00 lda #\$00			
03cc 8d a0 02 sta \$02a0			
03cf 4c 93 fc jmp \$fc93	ritorna ripristinando i vettori		
03d2 a9 08 lda #\$08	ROUTINE DI RICEZIONE BYTE		
03d4 85 a3 sta \$a3	contatore bit		
03d6 20 e5 03 jsr \$03e5	riceve bit		
03d9 26 bd rol \$bd	mette bit ricevuto in \$bd		
03db ea nop			
03dc ea nop			
03dd ea nop			
03de c6 a3 dec \$a3	ripete per 8 volte		
03e0 d0 f4 bne \$03d6			
03e2 a5 bd lda \$bd	ritorna con byte in accumulatore		
03e4 60 rts			
03e5 a9 10 lda #\$10	ROUTINE DI RICEZIONE DI UN BIT DAL REGISTRATORE		
03e7 2c 0d dc bit \$dc0d	attende fronte dal tape		
03ea f0 fb beq \$03e7			
03ec ad 0d dd lda \$dd0d	legge registro di stato 6526		
03ef 8e 07 dd stx \$dd07	setta registro high timer B		
03f2 48 pha	salva accumulatore in stack		
03f3 a9 19 lda #\$19	fa ripartire il timer B		
03f5 8d 0f dd sta \$dd0f			
03f8 68 pla	ricarica a da stack		
03f9 4a lsr	sposta flag di stato timer B		
03fa 4a lsr	in carry		
03fb 60 rts			
1000 fa 03 a0 e1	codici di autorun		
1004 ad 05 03 lda \$0305			
1007 85 26 sta \$26			
1009 ad 0c 03 lda \$030c	carica i puntatori con i dati		
100c 85 63 sta \$63			
100e ad 13 03 lda \$0313	di inizio e fine load		
1011 85 62 sta \$62			
1013 ad 17 03 lda \$0317			
1016 85 27 sta \$27			
1018 20 61 03 jsr \$0361	routine di load turbo		
101b a0 1f ldy #\$1f	ripristina vettori modificati con tabella standard		
101d b9 30 fd lda \$fd30,y			
1020 99 14 03 sta \$0314,y			
1023 88 dey			
1024 10 f7 bpl \$101d			
1026 20 53 e4 jsr \$e453	ripristina vettori I/O		
1029 ea nop			
102a a2 03 ldx #\$03	carica lo stack con i bytes nelle locazioni \$02ba (\$1000)		
102c bd ba 02 lda \$02ba,x			
102f 48 pha			
1030 ca dex			
1031 10 f9 bpl \$102c			
1033 a2 40 ldx #\$40	cancela programma fino all'indirizzo		
1035 a9 00 lda #\$00	di questa istruzione		
1037 9d b0 02 sta \$02b0,x			
103a ca dex			
103b 10 fa bpl \$1037			
103d a6 26 ldx \$26	setta puntatori di fine programma		
103f a4 27 ldy \$27			
1041 86 ae stx \$ae			
1043 84 af sty \$af			
1045 60 rts	termina ritornando al S.O.		
1046 be 02 be 02 7c 01 14 a7			
104e e4 a7 86 ae 28 62 00 32			
1056 4c 48 b2 30 2c f9 57 08			
105e e2 fc 4a f3 91 f2 0e f2			
1066 50 f2 33 f3 fb 03 fb 03			
106e b1 f1 fb			

ziando le principali differenze e le particolarità.

Possiamo notare che, oltre ai vettori di fine programma, c'è il settaggio anche di quelli di inizio, listato da \$1004 a \$1013.

C'è poi un diverso modo di attivare il programma che non sfrutta più il buffer della tastiera, bensì un accorgimento molto particolare che analizziamo insieme.

A proposito del settaggio dei

puntatori, possiamo solo dire che le operazioni da eseguire sono le medesime che abbiamo visto con il caricatore 1, vedi la modifica relativa all'istruzione LDA [indirizzo] ecc.; in più dobbiamo modificare, oltre ai vettori di fine Load (\$030c e \$0313), anche quelli di inizio Load (\$0305 e \$0317), prelevando i relativi dati dalle locazioni \$1052 - \$1059 - 104B - 105D. Il listato deve quindi diventare:

```
LDA #$ [dato in $104B]
NOP
STA $26
LDA #$ [dato in $1052]
NOP
STA $63
LDA #$ [dato in $1059]
NOP
STA $62
LDA #$ [dato in $105D]
NOP
STA $27
```

Analizziamo ora questo listato cercando di capire come viene attivato il programma caricato, una

volta terminato il Load.

Al ritorno dalle routine di Load Turbo notiamo, all'indirizzo \$101B, il ripristino dei vettori di sistema modificati per ottenere l'attivazione del caricatore, quindi c'è uno strano caricamento dello stack con dati letti dalle locazioni \$1000/\$1001/\$1002/\$1003 (\$1000 corrisponde a \$02BA).

Segue poi la cancellazione di un'area di memoria che inizia a \$02B0 e termina all'istruzione precedente a quella in esecuzione (\$1035) e, dopo il settaggio dei puntatori di fine programma \$AE/\$AF, termina il tutto con un RTS ossia un Return.

Ma come avviene l'Autorun? La risposta la troviamo nei dati memorizzati nello stack; infatti quando il processore incontra un'istruzione RTS preleva dallo stack gli ultimi due bytes memorizzati mettendoli nel Program Counter PC incrementando il byte basso di 1 (nel nostro caso trova \$FA e \$03).

Siccome il processore legge le istruzioni da eseguire all'indirizzo puntato dal program counter, proseguirà, dopo le operazioni viste, al nuovo indirizzo che è \$03FB; lì troverà di nuovo un RTS, vedi il listato, per cui ripeterà le operazioni viste prima prelevando dallo stack la seconda copia di bytes, che sono \$A0 e \$E1 corrispondenti all'indirizzo \$E1A1 (routine di RUN).

Una volta capito il trucco risulta semplice risolvere il problema; i sistemi utilizzabili sono molti, ma il più pratico penso sia quello utilizzante la nostra routine Dsave.

È sufficiente sostituire a \$FC93 (sul listato a \$03CF) l'indirizzo di salto \$0410 alla routine Dsave, cambiando l'istruzione JMP \$FC93 in JMP \$0410.

Ripetiamo poi le procedure viste con il caricatore 1: caricamento della routine Dsave, scrittura del nome del file, trasferimento della memoria da \$1000 a \$02BA dopo la modifica, ecc.

Una volta eseguite tutte queste operazioni possiamo attivare il

Listato 3. Caricatore 3

```

033c 03 02 03 05 03 52 4f 4c'.....rol
0344 4c 45 52 20 20 20 20 20'ler
034c 20 20 20 20 20 20 00 00' ..
0354 a9 10 lda #$10 ROUTINE DI RICEZIONE DI UN BIT DA REGISTRATORE
0356 2c 0d dc bit $dc0d attende il fronte dal tape
0359 f0 fb beq $0356
035b 8e 07 dd stx $dd07 memorizza nuovamente registro high timer B
035e a9 19 lda #$19 fa ripartire il timer B
0360 8d 0f dd sta $dd0f
0363 ad 0d dd lda $dd0d legge registro di stato 6526 e mette
0366 4a lsr flag di stato timer B
0367 4a lsr in carry bit
0368 60 rts
0369 a0 00 ldy #$00
036b 84 c0 sty $c0 start motore tape
036d a9 0b lda #$0b disabilita video
036f 8d 11 d0 sta $d011
0372 ca dex
0373 d0 fd bne $0372
0375 88 dey
0376 d0 fa bne $0372 ciclo di ritardo e azzeramento registri
0378 78 sei
0379 60 rts
037a a9 08 lda #$08 ROUTINE DI RICEZIONE DI UN BYTE
037c 85 a3 sta $a3 memorizza vettore di conteggio bit
037e 20 54 03 jsr $0354 riceve il bit di ritorno nel carry
0381 26 bd rol $bd viene messo in $BD tramite uno shift
0383 ee 20 d0 inc $d020 cambia colore sfondo
0386 c6 a3 dec $a3 decrementa contatore bit
0388 d0 f4 bne $037e ripete fino a 8
038a a5 bd lda $bd carica in accumulatore il byte ricevuto
038c 60 rts
038d 20 69 03 jsr $0369 start registratore e off video
0390 a9 07 lda #$07 setta registro low timer B
0392 8d 06 dd sta $dd06 per un tempo di conteggio di 263 microS
0395 a2 01 ldx #$01
0397 20 54 03 jsr $0354
039a 26 bd rol $bd
039c a5 bd lda $bd
039e c9 02 cmp #$02 SINCRONIZZAZIONE
03a0 d0 f5 bne $0397
03a2 a0 09 ldy #$09 tra
03a4 20 7a 03 jsr $037a
03a7 c9 02 cmp #$02
03a9 f0 f9 beq $03a4 COMPUTER e TAPE
03ab c4 bd cpy $bd
03ad d0 e8 bne $0397
03af 20 7a 03 jsr $037a
03b2 88 dey
03b3 d0 f6 bne $03ab
03b5 60 rts
03b6 58 cli INIZIO PROGRAMMA
03b7 20 8d 03 jsr $038d preparazione e sincronizzazione
03ba 85 c1 sta $c1 riporta indirizzo low inizio programma
03bc 20 7a 03 jsr $037a caricamento da tape dell'indirizzo
03bf 85 c2 sta $c2 high di inizio programma
03c1 20 7a 03 jsr $037a
03c4 85 2d sta $2d indirizzo low fine programma
03c6 20 7a 03 jsr $037a
03c9 85 2e sta $2e indirizzo high fine programma
03cb 20 7a 03 jsr $037a
03ce 48 pha codici di indirizzamento che
03cf 20 7a 03 jsr $037a vengono memorizzati nello stack
03d2 48 pha in modo di ottenere l'attivazione
03d3 98 tya programma (autorun)
03d4 48 pha
03d5 20 7a 03 jsr $037a RICEZIONE DI UN BYTE PROGRAMMA
03d8 91 c1 sta ($c1),y MEMORIZZAZIONE IN RAM
03da e6 c1 inc $c1
03dc d0 02 bne $03e0
03de e6 c2 inc $c2 incremento dei puntatori
03e0 a5 c1 lda $c1 di memorizzazione del
03e2 c5 2d cmp $2d programma in caricamento
03e4 a5 c2 lda $c2
03e6 e5 2e sbc $2e
03e8 90 eb bcc $03d5
03ea e6 c0 inc $c0 al termine ferma il motore del registr.
03ec 8c a0 02 sty $02a0
03ef 20 93 fc jsr $fc93 ripristino dei vettori
03f2 20 a0 e5 jsr $e5a0 di I/O
03f5 40 rti ritorno al S.O.
03f6 20 59 a6 jsr $a659 routine di
03f9 4c ea a7 jmp $a7ea RUN
03fc 00 brk

```

1000 b6 03 b5 00 00 00 00'..... \$03b6 indirizzo di start caricatore

Listato 4. Caricatore 4

0351 a0 08	ldy #08	routine di ricezione byte da nastro	03d1 20 51 03	jsr \$0351	carica byte di verifica
0353 a9 10	lda #10		03d4 45 90	eor \$90	
0355 2c 0d dc	bit \$dc0d	attende fronte da tape	03d6 f0 03	beq \$03db	se ok esegue run
0358 f0 fb	beq \$0355		03d8 4c e2 fc	jmp \$fce2	altrimenti reset
035a ad 0d dd	lda \$dd0d	legge registro di stato 6526	03db 85 90	sta \$90	
035d 8e 0f dd	stx \$dd0f	fa ripartire timer B	03dd a9 27	lda #27	ferma motore
0360 4a	lsr	sposta flag timer B in carry	03df 85 01	sta \$01	
0361 4a	lsr		03e1 c8	iny	
0362 66 bd	ror \$bd	mette bit da carry in \$bd	03e2 84 c0	sty \$c0	
0364 ee 20 d0	inc \$d020	incrementa colore bordo	03e4 a6 ae	ldx \$ae	setta puntatori di fine programma
0367 88	dey	ripete per ricevere 8 bit	03e6 a4 af	ldy \$af	esce al S.O.
0368 d0 e9	bne \$0353		03e8 58	cli	
036a a5 bd	lda \$bd	carica accumulatore con byte ricevuto	03e9 18	clc	
			03ea 60	rts	
036c 60	rts		03eb 4c 6d 03	jmp \$036d	
036d 78	sei	punto di entrata			
036e a9 07	lda #07	fa partire motore registratore	1000 20 53 e4	jsr \$e453	inizializza i vettori
			1003 20 44 e5	jsr \$e544	cancella il video
0370 85 01	sta \$01		1006 a0 01	ldy #01	
0372 a2 19	ldx #19		1008 8c 21 d0	sty \$d021	colore fondo bianco
0374 a9 80	lda #80		100b c8	iny	
0376 8d 06 dd	sta \$dd06	setta registro low timer B	100c 8c 86 02	sty \$0286	colore testo rosso
0379 a9 01	lda #01		100f a2 02	ldx #02	
037b 8d 07 dd	sta \$dd07	setta registro high timer B	1011 a0 02	ldy #02	
037e 8e 0f dd	stx \$dd0f	fa partire timer B	1013 18	clc	
0381 a0 09	ldy #09		1014 20 f0 ff	jsr \$fff0	posiziona il cursore 2a riga 2a colonna
0383 20 53 03	jsr \$0353				
0386 c9 02	cmp #02		1017 a0 ef	ldy #\$ef	
0388 d0 f7	bne \$0381	sincronismo	1019 b9 00 02	lda \$0200,y	stampa la scritta memorizzata in \$1048
038a 20 51 03	jsr \$0351				routine di stampa
038d c9 02	cmp #02	tra	101c 20 d2 ff	jsr \$ffd2	
038f f0 f9	beq \$038a		101f c8	iny	
0391 c9 52	cmp #52	registratore	1020 d0 f7	bne \$1019	
0393 d0 ec	bne \$0381	e	1022 b9 41 03	lda \$0341,y	stampa nome file prelevato dal buffer del registratore
0395 20 51 03	jsr \$0351	computer			
0398 c9 42	cmp #42		1025 20 d2 ff	jsr \$ffd2	
039a d0 e5	bne \$0381		1028 c8	iny	
039c 20 51 03	jsr \$0351	riceve gli indirizzi di inizio load	1029 c0 10	cpy #10	
039f 85 ac	sta \$ac	low	102b d0 f5	bne \$1022	
03a1 85 c1	sta \$c1		102d a9 01	lda #01	colore testo = colore sfondo
03a3 20 51 03	jsr \$0351				
03a6 85 ad	sta \$ad	high	102f 8d 86 02	sta \$0286	
03a8 85 c2	sta \$c2		1032 a2 6d	ldx #\$6d	inserisce l'indirizzo di inizio load turbo nei puntatori al load di sistema
03aa 20 51 03	jsr \$0351	e di fine load			indirizzo loader turbo
03ad 85 ae	sta \$ae	low	1034 a0 03	ldy #03	
03af 20 51 03	jsr \$0351				
03b2 85 af	sta \$af	high	1036 8e 30 03	stx \$0330	
03b4 84 90	sty \$90		\$036d		
03b6 20 51 03	jsr \$0351	carica byte programma	1039 8c 31 03	sty \$0331	mette il codice del comando load/run nel primo byte del buffer della tastiera
03b9 91 ac	sta (\$ac),y	memorizza in ram	103c a9 83	lda #\$83	e mette 1 nel puntatore del numero di caratteri nel buffer di tastiera
03bb 45 90	eor \$90				reset caldo
03bd 85 90	sta \$90		103e 8d 77 02	sta \$0277	'nome pro
03bf e6 ac	inc \$ac				'gramma :
03c1 d0 02	bne \$03c5		1041 a9 01	lda #01	'1
03c3 e6 ad	inc \$ad	incrementa			
03c5 a5 ac	lda \$ac		1043 85 c6	sta \$c6	
03c7 c5 ae	cmp \$ae	i			
03c9 d0 eb	bne \$03b6		1045 4c 7b e3	jmp \$e37b	
03cb a5 ad	lda \$ad	puntatori	1048 4e 4f 4d 45 20 50 52 4f		
03cd c5 af	cmp \$af		1050 47 52 41 4d 4d 41 20 3a		
03cf d0 e5	bne \$03b6		1058 20 8b e3 a7		

lizzare la routine DSAVE modificando l'indirizzo \$FCE2 in \$0410 ed eseguendo le stesse operazioni usate per gli altri caricatori.

Alcuni consigli

Certi programmi sono divisi in due parti che vengono caricate in successione, normalmente la prima carica la seconda tramite un semplice Load e per eliminare l'effetto di questo Load che agisce solo su nastro è consigliabile usare il seguente sistema:

- Caricare il primo programma e listarlo, di solito appare solo una linea del tipo 10 SYS 2064: LOAD.

- Siccome non è possibile modificare il Load scrivendo il nome del file e il numero del dispositivo, tanto per intenderci in questo modo: 10 SYS 2064:LOAD" c",8, perché si sovrascriverebbe una parte del programma assembler, è meglio cancellare il Load, risalvare il programma, sempre usando lo Zoom, e creare un caricatore come il seguente. Supponiamo che il primo programma si chiami PIPPO e il secondo P: questo programma funziona scrivendo sullo schermo i comandi di Load e di Run dei due file, attivandoli e inserendo nel buffer della tastiera i caratteri Home e quattro Return, uno per ogni comando.

- Può succedere poi, durante il caricamento da nastro, che un tentativo di Load sia infruttuoso a causa della testina del registratore starata o sporca.

Provate a caricare più volte il programma e quando siete sicuri del buon funzionamento del registratore ripetete le operazioni di trasferimento.

Speriamo di aver dato una mano a risolvere i problemi di chi, pur avendo un comodo drive, deve ancora affannarsi con il registratore per giocare con il game preferito o che, per copiare un floppy, deve paradossalmente caricare dal nastro l'utility.

Marco Moioli

Una simulazione davvero ben riuscita, ricca di varianti e molto curata per divertenti partite nel tentativo di eguagliare i primati del mitico barone rosso.



Flying Shark II

Dovete sapere che, durante la seconda guerra mondiale, gli inglesi mandarono in ricognizione in zone sperdute dell'Africa centrale una squadriglia di caccia adeguatamente equipaggiata, con il compito di fotografare una base segreta tedesca.

Partirono il 23 luglio 1943, sorvolarono la Manica, la Francia, l'Italia e il Mediterraneo. Proprio sopra questo mare furono individuati dai radar tedeschi. Uno stormo di stuka si levò subito in volo per intercettare la squadriglia. Appena sopra l'Africa, ci fu lo scontro: fu violentis-

simo e le perdite da ambo le parti furono ingenti. Alla lunga però, la più preparata squadriglia tedesca cominciò ad avere la meglio sopra quella inglese colta quasi di sorpresa. Il massacro fu totale, non venne risparmiato nessuno, nemmeno i pochi che si erano lanciati con il paracadute. Solo il più piccolo aereo inglese, equipaggiato con l'attrezzatura fotografica, riuscì a sfuggire alla ferocia tedesca volando a bassa quota. Proseguì la sua rotta fino a destinazione, scattò le foto ma fu nuovamente avvistato e la contraerea cercò di abbatterlo. Compiuta la sua missione, il pilota inglese decise di tornare indietro, ma mentre stava virando fu colpito a un'ala. Costretto all'atterraggio anche per mancanza di carburante, trovò fortunatamente una pista abbandonata. Mimetizzò l'aereo e, giunta la notte, si riposò. La mattina seguente cercò negli hangar abbandonati il necessario per riparare e rifornire il velivolo. Trovato tutto il materiale, si mise subito a lavoro e nel pomeriggio concluse la riparazione. Portò l'aereo sulla pista ma proprio quando stava per decollare fu assalito dai tedeschi che lo catturarono e lo rinchiusero assieme agli altri prigionieri.

Intanto in Inghilterra si dava la missione per fallita in quanto si sapeva solo dell'intercettazione della squadriglia da parte degli aerei tedeschi. Qualche giorno più tardi, l'aeroporto di Manchester captò su una frequenza radio militare il



messaggio in codice del pilota che intanto era riuscito a evadere e si accingeva al decollo: "Sono salvo, sto partendo da un aeroporto abbandonato, venitemi incontro, aiuto!". Subito un'altra squadriglia di aerei fu fatta decollare per proteggere il pilota e le preziose fotografie. Il messaggio fu però intercettato dai tedeschi...

Come si gioca

Il gioco, come avrete già capito, consiste nel riportare sano e salvo il velivolo in Inghilterra. Munito solo di una debole mitraglietta, il pilota dovrà tener testa ai tedeschi che cercheranno in tutti i modi di ostacolare la sua avanzata. L'aereo partirà dalla pista abbandonata e dovrà atterrare in Inghilterra. Durante il viaggio sorvolerà il Mediterraneo e numerose basi nemiche dove carri armati, contraerea, aerei isolati o in formazione lo attaccheranno senza un attimo di tregua. Per ritornare in patria avete tre vite a disposizione e per giocare dovete inserire in porta due il joystick. La presentazione è molto ben disegnata: è composta da due illustrazioni, la prima si crea dal basso verso l'alto, mentre la seconda è un tramonto sul mare e appare ogni volta che si finisce una partita. Lo scrolling del video è del tipo verticale, dall'alto verso il basso. Lo sfondo è decisamente indovinato anche se forse un po' monotono, mentre gli sprite sono ben disegnati. Il piccolo aeroplano da pilotare è curato in tutti i dettagli, l'elica, i colori, il cambiamento della forma quando gira...

Ogni avversario utilizza un particolare tipo di sparo: a terra, fermo o in movimento che sia, spara solo in orizzontale o solo in verticale o in entrambi i modi (solo le basi nere si distinguono). Nelle diagonali sparano solo gli aerei rossi e arancio, che infatti sparano nella direzione in cui si muovono e compiono un percorso molto bizzarro sullo schermo. Gli aerei arancio sono più facili da abbattere e sparano di meno: quelli rossi, anche se non girano mai in formazione ma isolati, vi daranno qualche problema in più. Per distruggere i carri armati sono necessari più colpi. Sul tracciato vi sono anche delle jeep o degli aerei fermi: se li distruggete guadagnerete una notevole quantità di punti, necessari per arrivare al punteggio bonus di diecimila punti con il quale si vince una vita. Sorvolando il mare o gli stretti, vi attaccheranno anche alcuni caccia-

torpedinieri, ma in modo così poco aggressivo da non costituire un grande problema. Ogni tanto lo scrolling si fermerà: state molto all'erta, perché i nemici appariranno all'improvviso. Il programma riesce a mantenere sullo schermo una notevole quantità di sprite, circa dieci o anche più, a scapito però della più bassa velocità di azione. Il punto dolente del gioco è costituito dalle opzioni purtroppo misere (è infatti possibile giocare solo una alla volta); non vi è un brano musicale di sottofondo e il sonoro è costituito solo dagli effetti particolari degli spari. In fondo allo schermo sono visualizzati il punteggio e gli aeroplani ancora a disposizione. La difficoltà non è eccessiva e il gioco è quindi adatto a tutti, abili e meno abili.



Consigli e conclusioni

I punti più difficili sono senz'altro costituiti da quelli in cui si sorvola il mare aperto. I colpi sparati dagli aerei nemici sono così piccoli che risultano quasi invisibili. Vi consigliamo di seguire il movimento del nemico come un'ombra, oppure di stargli lontano se conoscete il suo percorso di volo.

Non cercate di colpire a tutti i costi postazioni a terra che per esempio sparano in verticale, potrebbe costarvi molto caro: infatti, malgrado la difficoltà non insormontabile, il giochino presenta un tragitto molto lungo, quindi è meglio economizzare al massimo le vite a disposizione. Con un po' di pratica anche questi pochi problemi si risolveranno!

Oscar Maeran



Due routine per migliorare le prestazioni del Dos; una mini espansione dell'editor; una routine per controllare il drive e un freezer per concedere una pausa al vostro C64.

Trucchi e scorciatoie

• Loader

Il sistema operativo del C64 ha, come è noto, molte lacune. Tra queste c'è l'impossibilità di caricare un file di dati in una zona prefissata della memoria. Infatti quando si salva un programma Basic, o più in generale un file programma, il sistema operativo provvede automaticamente a memorizzare su disco la zona di provenienza dei dati (\$0801 per i programmi Basic e l'indirizzo di allocazione per le routine in linguaggio macchina).

L'indirizzo salvato su disco viene quindi utilizzato per stabilire da quale locazione deve iniziare la

re dall'indirizzo memorizzato dal Dos su disco) oppure a partire dall'inizio dell'area Basic. Queste due diverse modalità di caricamento corrispondono rispettivamente alle istruzioni Load"nomefile",8,1 e Load"nomefile",8.

Ovviamente la situazione non si presenta particolarmente tragica se si ha a che fare unicamente con programmi Basic. Tuttavia in alcuni casi può diventare veramente insopportabile. Ad esempio capita spesso di dover caricare una routine in linguaggio macchina, una schermata di alta risoluzione o più semplicemente una serie di dati salvati sotto forma di file programma, in un'area di memoria diversa da quella originaria, cioè quella fissata al momento del salvataggio. In questo caso purtroppo si deve ricorrere a un monitor per rilocare manualmente quanto

sta breve routine, interamente in linguaggio macchina, permette infatti di caricare, a partire da un indirizzo qualsiasi, un file programma memorizzato su disco o nastro. Per utilizzare la routine copiate il **listato 1** e quindi digitate Run.

Per attivare la routine dovete digitare: Sys s, "nomefile", dvn, indirizzo. S è la variabile che contiene l'indirizzo di allocazione di Loader (vedere linea 100), dvn indica il numero associato alla periferica su cui risiede il file da caricare e indirizzo è l'indirizzo secondario, da esprimere in notazione decimale, da cui deve partire il caricamento. Tenete presente che la routine può essere rilocata modificando il valore di S alla linea 100.

• Saver

Il problema opposto rispetto a quello visto in precedenza consiste nella impossibilità di salvare pezzi di memoria. Questa operazione si rivela particolarmente

Listato 1. Loader

```
100 s=700:print"[clr]":rem cambiare s per rilocare la routine
150 ford=s to s+63:ready:poked,y:next
200 data 234,32,253,174,32,158,173,32
210 data 163,182,32,189,255,32,253,174
220 data 32,138,173,32,247,183,162,8
230 data 165,20,160,0,32,186,255,32
240 data 253,174,32,138,173,32,247,183
250 data 169,0,166,20,164,21,32,213
260 data 255,165,157,240,9,169,13,32
270 data 210,255,152,32,205,189,96,234
280 print "per attivare:
290 print:
    print"sys"s"[left],nomefile,dvn,indirizzo"
```

memorizzazione dei dati letti. Il programmatore è in grado di scegliere solo se caricare i dati nella zona di provenienza (cioè a parti-

modo più efficiente potete utilizzare Loader. Que-

è stato caricato. Se però volete risolvere il problema in

Listato 2. Saver

```
100 s=8305:print"[clr]":rem cambiare s per rilocare
110 ford=s to s+61:ready:poked,y:next
120 data 234,32,253
130 data 174,32,158,173,32,163,182,32
140 data 189,255,32,253,174,32,138,173
150 data 32,247,183,160,1,166,20,169
160 data 1,32,186,255,32,253,174,32
170 data 138,173,32,247,183,132,251,133
180 data 252,32,253,174,32,138,173,32
190 data 247,183,168,166,20,169,251,76
200 data 216,255,234
300 print"per attivare:"
310 print:print"sys"s"[left],nomefile,dvn,init,fine+1"
```


utile quando si vuole memorizzare una schermata di alta risoluzione, i dati di alcuni sprite o routine in linguaggio macchina. Per porre rimedio a questa situazione potete servirvi di *Saver*. La routine è scritta interamente in linguaggio macchina e per utilizzarla dovete copiare il **listato 2** e quindi digitare Run. La routine si attiva con: Sys s, "nomefile", dvn, init, fine+1. I primi tre parametri

tosio oppure il drive ha dei problemi di allineamento. Se vi trovate in questa condizione potete risolvere il dilemma con *Disk doctor*. Questa mini utility, scritta interamente in Basic, è in grado di effettuare la verifica delle capacità della testina del drive entro margini di errore molto buoni. La routine fa compiere una lettura del primo blocco delle tracce 1, 35 e 15. Se il drive si trova in perfette condizioni la lettura delle tracce non deve causare nessun errore, cioè la spia rossa non deve lampeggiare al termine dell'operazione.

bisogno di un interruttore magico per congelare istantaneamente ogni attività del C64 e permettervi di assentarvi per tutto il tempo necessario. Se sentite la mancanza di un tale strumento allora *Time out* fa proprio per voi. Questa utility vi permette di bloccare, con la semplice pressione di un tasto, ogni attività del C64 per tutto il tempo che volete. Per utilizzare *Time out* copiate il **listato 4** e date il consueto Run. Ogni volta che volete congelare le attività del C64 dovete premere contemporaneamente i tasti CTRL e Commodore. Per riattivare il computer è invece sufficiente digitare un tasto qualsiasi (a eccezione dei tasti CTRL Commodore e Run/Stop). Tenete presente che la routine funziona manipolando il puntatore alla routine di interrupt e quindi per funzionare è necessario che l'eventuale programma in memoria non modifichi questo puntatore.

Listato 3. Disk doctor

```
10 t=1:goto50
20 ift=18thenend
30 ift=1thent=35:goto50
40 ift=35thent=18
50 open15,8,15
60 open2,8,2,"#"
70 print#15,"ul,2"0;t,1
80 print"track"t
90 close2:close15
100 goto20
```

Listato 4. Time out

```
10 x=700:rem cambiare x per rilocare
20 forj=xtox+36:reada:pokej,a:next
30 x1=int(x/256):pokej+1,x+13-(x1*256):pokej+3,x1:end
40 data169,201,162,2,120,141,20,3,142,21,3,88,96,165,203,201
50 data 3:rem codice tasto di attivazione
60 data 208,15,32,159,255,32,228,255,208,7
70 data 173,0,220,201,127,240,241,76,49,234
```

hanno lo stesso significato degli omonimi parametri visti per la routine precedente. Gli ultimi due parametri sono rispettivamente l'indirizzo di inizio e quello finale aumentato di 1 del segmento di memoria da memorizzare. La routine può essere rilocata per adattarla alle vostre esigenze semplicemente modificando il valore della variabile S alla linea 100.

• Disk doctor

Quante volte vi sarà capitato di vedere lampeggiare la spia del drive senza motivo apparente. Le possibili cause di una tale situazione possono essere di vario tipo, prima fra tutte l'assenza del disco nel drive. Se si esclude questa possibilità si hanno soltanto due alternative: il disco nel drive è difet-

ne. Se invece al termine dell'operazione la spia rossa lampeggia oppure se il drive fa fatica a leggere, significa che la testina deve essere riallineata. Ovviamente il test effettuato deve essere inteso soltanto come indicazione di massima e non come un responso definitivo. Per utilizzare *Disk doctor* dovete copiare il **listato 3** e quindi digitare Run per iniziare la verifica.

• Time out

Vi sarà sicuramente capitato molto spesso di essere interrotti nel mezzo di una avvincente partita col vostro videogame preferito oppure dopo aver appena lanciato un programma. In situazioni di questo tipo si sente proprio

• Screen clear

Fra le numerose facilitazioni messe a disposizione dall'editor del Commodore 128 c'è la possibilità di cancellare la porzione dello schermo a partire dalla posizione del cursore fino alla fine dello schermo con la semplice pressione di un tasto. Questa caratteristica si rivela utile in moltissime situazioni, prima fra tutte la stesura dei programmi. *Screen clear* implementa questa opzione nell'editor del C64. Per utilizzare la routine dovete copiare il **listato 5** e quindi dare il solito Run. *Screen clear* si attiva con Sys 679 mentre con Sys 706 si disattiva. Per cancellare la porzione di schermo dalla linea in cui si trova il cursore fino alla fine dello schermo basta premere il tasto £. Tenete presente che la routine non può essere rilocata quindi non tentate di modificare il valore della variabile i alla linea 100.

Daniele Maggio

Listato 5. Screen clear

```
100 print"[clr]":fori=679to760:reada:c=c+a:pokei,a:next
110 ifc<10783thenprint"[clr]errore[32]nei[32]data":end
120 print"screen[32]clear[32]installato"
130 print:print"sys[32]679[32]attiva"
140 print:print"+[32][32]cancella"
150 print:print"sys[32]706[32]disattiva"
160 end
170 data 173,20,3,174,21,3,224,2,240,16,133,251,134,252,120,169,211,162,2,141
180 data 20,3,142,21,3,88,96,120,165,251,166,252,224,234,208,7,141,20,3,142,21
190 data 3,88,96,165,197,201,48,240,3,108,251,0,165,157,240,249,166,214,32,255
200 data 233,232,224,25,208,248,169,1,133,204,160,0,166,214,24,32,240
210 data 255,108,251,0
```




Torri, guardiani, casseforti, terminali e missili: sono i protagonisti della nostra caccia al perfido Elvin, che vuole distruggere il mondo. Ci riuscirà?

Elvin: missione impossibile

Se pensavate che il vostro scontro con il perfido Elvin fosse finito al termine di Missione Impossibile quando, sgomento, vi guardava entrare nella sua stanza e capiva che avevate risolto il gioco, avete completamente sbagliato; Elvin è vivo, vegeto e ancora libero come un fringuello.

Anche questa volta ha minacciato di distruggere il mondo e la vostra agenzia vi ha incaricato di catturare il perfido pazzo, visto che già una volta siete riusciti nell'intento.

La missione è semplice e concisa, dovette introdurvi all'interno della roccaforte di Elvin, riunire i codici di accesso formati da tre numeri, per ogni torre, affrontare i robot guardiani, trovare le casseforti e recuperare le sequenze musicali, legare le sequenze musicali per formare la melodia segreta che aprirà le porte dell'ascensore che porta direttamente nella stanza di controllo di Elvin e, per finire, dovete scoprire quale tra i terminali presenti nella stanza disattiva i codici di controllo per la partenza dei missili.

Come vedete non c'è di che annoiarsi; riuscirete a salvare il mondo? Tutti sperano di sì.

Il gioco

Mentre il programma viene caricato, vedrete la roccaforte di Elvin dall'esterno: si tratta di una torre centrale (che è la stanza di controllo dello squilibrato) circondata da altre otto torri (in cui si svolge tutto il gioco).

Quando il gioco ha inizio vi troverete in una di queste torri e, per poter passare a una torre successiva, dovrete risolvere il primo enigma, cioè trovare i numeri che formano la combinazione di apertura delle porte della torre seguente; quando avrete trovato i numeri dovrete usare il computer tascabile per formare la sequenza esatta.

Attivate il computer tascabile premendo il pulsante di Fire (potete attivarlo solamente quando sarete in un corridoio o nell'ascensore): apparirà una manina che vi servirà per le scelte, portatela sull'icona che indica i tre numeri e premete Fire, al posto della mappa della roccaforte verranno visualizzate tre finestre, fate scorrere i numeri con i tasti comando fino a quando sotto al numero non comparirà la parola Found e a questo punto passate al numero successivo; quando sotto lo schermo apparirà Complete avrete aper-

to le porte della torre adiacente. Nelle torri dovete anche scovare le casseforti e trovare le sequenze musicali; prima di lasciare la torre dovete registrare la sequenza trovata.

Dopo aver attivato il computer tascabile selezionate il registratore e registrate la sequenza trovata, poi ascoltate le sequenze (avete un contatore digitale che vi aiuterà a capire dove inizia e dove finisce una sequenza).

Quando avrete suonato sei sequenze complete potrete entrare nella torre di controllo.

Ricordate di registrare subito i brani trovati, perché se uscirete dalla torre senza averlo fatto perderete la sequenza e non sarete più in grado di risolvere il gioco.

All'interno di ogni stanza dovete cercare in ogni oggetto visibile (dalla scrivania al cestino dei rifiuti), non tralasciate neppure gli oggetti d'arredamento come i quadri o le piante. Una finestrella si attiverà a fianco dell'agente, comparirà la parola Searching e una barra indicherà il tempo necessario per la ricerca.

Dovete tenere il joystick premuto in avanti finché non scomparirà la barra; se dovete necessariamente abbandonare la ricerca (per esempio nel caso che un robot punti diritto verso di voi) non preoccupatevi, potrete riprendere la ricerca dove l'avevate lasciata (sempre che non usciate dalla stanza, in questo caso al vostro ritorno incomincerete tutto dall'inizio).

Una volta terminato il controllo l'oggetto scomparirà; le risposte che potrete ottenere dalla ricerca sono quattro e precisamente:

- La parola nothing (niente). Essa indica che avete solamente perso tempo e rischiato la vita.
- Un numero di codice.
- Una icona di comando dei terminali di sicurezza.
- Una estensione di tempo.

In ogni stanza della roccaforte di Elvin potrete usare le icone di comando; vicino all'ingresso troverete i terminali di sicurezza e potrete usare le icone in vostro possesso.

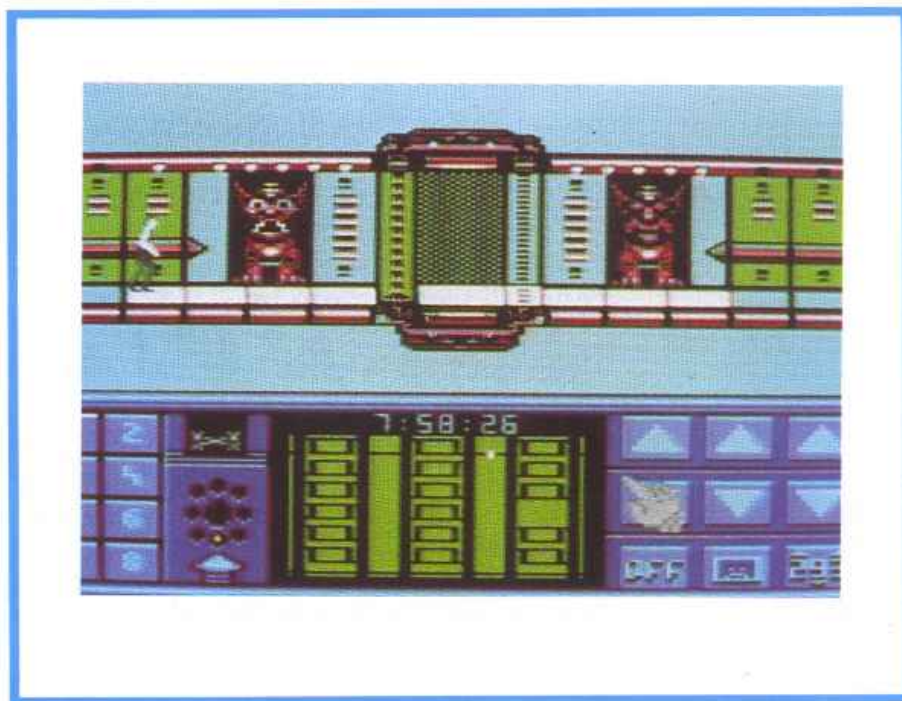
Per attivare il terminale mettetevi di fronte ad esso e spingete la barra del joystick in avanti quando sarà comparso lo schermo e potrete quindi utilizzare le vostre icone:

- **Piattaforma.** Troverete due icone che rappresentano le piattaforme, una serve per muovere quelle verticali, l'altra quelle con movimento orizzontale.

- **Presa elettrica.** Disattiva temporaneamente i robot presenti nella stanza.

- **Lampadina.** Troverete alcune stanze avvolte nell'oscurità più totale e questa icona vi permetterà di attivare l'impianto di illuminazione.

- **Bomba.** Innesca una bomba a orologeria che dovete collocare per far saltare la cassaforte. Quando vi trovate davanti alla cassaforte tirate indietro il joy-



stick e premete il fuoco per innescare la bomba che esploderà dopo alcuni secondi, mettetevi al riparo e tornate alla cassaforte successivamente per cercare le sequenze musicali.

- **Mina.** Innesca una mina che dovete collocare, ma fate attenzione perché esploderà al primo contatto e, se inavvertitamente la urterete, cadrete nella trappola da voi stessi preparata. L'esplosione apre un grosso buco nel pavimento che potrete superare solamente saltando.

Dovete affrontare diversi robot nel vostro vagabondare per le stanze della roccaforte; alcuni sono letali, altri meno, però una cosa li accomuna: faranno di tutto per uccidervi. I robot che troverete hanno nomi fantasiosi, e cioè:

Pagina mancante (pubblicità)

Anche voi potrete realizzare videogames eccezionali con questi due fantastici programmi:

Shoot'em up Construction Kit e Game Maker.

Sono due completissimi generatori di

videogames che vi permetteranno di realizzare qualsiasi gioco abbiate in mente senza richiedere la stesura di una sola riga di programma.



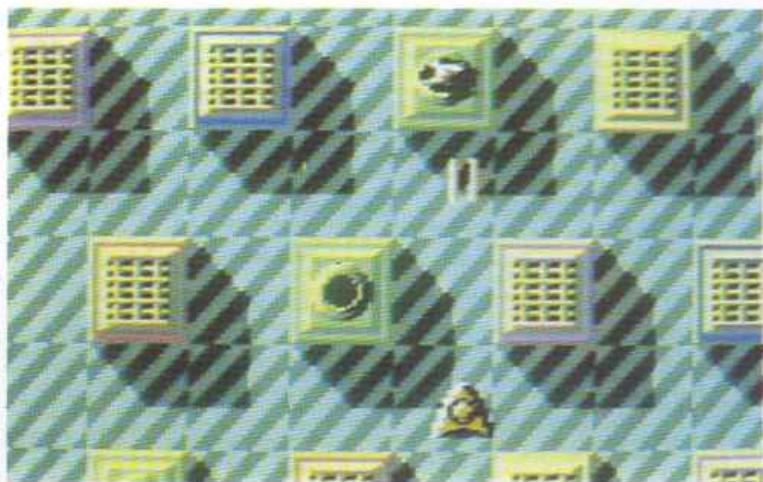
Shoot'em up construction kit

Fra le numerose espansioni proposte su questa rivista alcune hanno caratteristiche e qualità tali da renderle adatte alla creazione di videogame. La più importante e anche quella di maggior successo fra tutte è senza dubbio Basic Lightning, per cui è stato necessario, date le notevoli dimensioni e complessità, ricorrere a un corso a puntate per illustrarne tutte le caratteristiche con la necessaria accuratezza.

È inutile dire che anche con questo valido aiuto la realizzazione di un videogame resta comunque un'impresa ardua. Malgrado le numerosissime istruzioni rendono semplicissima e immediata la gestione dell'alta risoluzione e degli effetti sonori, resta sempre la fase di programmazione che, in ogni caso, non può essere in alcun modo agevolata se non si ha già

chiaramente in mente ciò che si vuole fare. Inoltre non si deve dimenticare che non tutti riescono a destreggiarsi con uguale successo, a livello di programmazione, nella creazione di immagini e di effetti sonori. Insomma ciò che manca veramente a questa espansione è una serie di tools pronti per l'uso che renda completamente automatica tutta la fase di definizione del videogame, dalle primissime fasi sino alla parte finale di assemblaggio. Per questo motivo sono stati sviluppati diversi programmi per la creazione au-

tomatica di videogame, cioè veri e propri programmi che permettono di costruire tutti gli elementi di un videogame e quindi di assemblarli senza richiedere da parte dell'utente nemmeno una riga di programma. Ci occuperemo quindi proprio dei due massimi esponenti di questa categoria:



Shoot'em up construction kit e Game maker. In termini di qualità, valore e capacità sono in assoluto eccezionali.

Le caratteristiche di questi programmi sono abbastanza diverse: il primo è orientato alla creazione di arcade, cioè di giochi del tipo spara e scappa con un unico e vasto scenario articolato in varie schermate che si susseguono automaticamente per scrolling (di questa categoria fanno parte, ad esempio, Sanxion, Star Wars e Nebulus) oppure in seguito al superamento del livello (come in Commando), mentre il secondo si presta meglio alla creazione dei

particolare; bastano infatti un drive 1541 o compatibile oppure un registratore, un joystick e un monitor, possibilmente a colori.

Entrambi i programmi sono facilmente reperibili sul mercato ma se non riuscite a trovarli potete rivolgervi direttamente alle ditte importatrici. In ogni caso però assicuratevi che il programma sia corredato dai demo poiché nel corso degli articoli faremo largo riferimento a essi. Più precisamente con Shoot'em up vengono forniti quattro giochi mentre con Game maker ne viene fornito uno.

In questo articolo cominciamo ad esaminare il primo dei due pro-

finché l'opzione indicata non appare lampeggiante e premete il tasto di Fire. Sempre seguendo la stessa procedura, nel sottomenù selezionate l'opzione Change device per specificare la periferica che possedete (quella attiva viene indicata sulla parte superiore del menù stesso) e quindi l'opzione Load.

Dal menù visualizzato selezionate il primo dei quattro giochi forniti (il nome di questi file termina con la specifica All data). Se state utilizzando il registratore l'operazione richiederà alcuni minuti in quanto devono essere caricati ben 246 blocchi. Al termine del caricamento selezionate l'opzione Exit, con cui si torna al menù principale, e quindi l'opzione Test Game. Quando compare il nuovo sottomenù selezionate Cheat mode: verrete immediatamente proiettati in un arcade interamente costruito con Shoot'em up construction kit, dagli sfondi a tutto ciò che si muove e si sente.

Come verrà spiegato successivamente, l'opzione Cheat mode permette di testare il videogame creato (oppure una sua parte) avendo a disposizione un numero infinito di vite. Questo vuol dire che, se volete, potrete visitare l'intero gioco per rendervi conto di persona delle potenzialità del programma. Potete fare ritorno al menù principale premendo la barra spaziatrice quando volete. Terminata la prova, ritornate al menù principale (sempre selezionando l'opzione Exit dal menù in cui vi trovate).

Il menù principale

La struttura del menù principale è predisposta in modo tale che l'ordine delle opzioni corrisponde esattamente all'ordine corretto delle operazioni da eseguire per creare un videogame (questo naturalmente se si inizia da zero). Se parte del lavoro è già stato fatto dovete saltare direttamente alle operazioni non ancora svolte. Ogni opzione del menù può essere selezionata con il joystick oppure

Figura 1. Menù principale di Shoot'em up construction kit.



videogame cosiddetti a stanze, cioè di giochi in cui l'azione è gestita quasi completamente dal giocatore che ha la facoltà di muoversi liberamente nell'insieme di stanze, cioè schermate, in cui è suddiviso l'intero scenario del gioco (alcuni esempi di questa categoria sono Builder Dash e Impossible Mission). Le caratteristiche comuni dei due costruttori di videogames sono la semplicità d'uso e la versatilità. Vi stupirete per la facilità e la velocità con cui riuscirete a costruire scenari, animazioni, effetti sonori e quindi ad assemblarli nel gioco finale.

Entrambi permettono di creare con estrema facilità una versione stand alone del videogame (cioè una versione del gioco che non ha bisogno del generatore per funzionare) che può essere liberamente commercializzata. Inoltre non richiedono un equipaggiamento

grammi, e cioè Shoot'em up Construction Kit. Per caricare questo programma dovete digitare in modo diretto l'usuale comando Load nella forma richiesta dal particolare supporto impiegato.

Se possedete un registratore dovete quindi digitare Load"shoot'em up" e poi premere Return; se invece avete un drive dovete digitare Load"shoot'em up",8,1 e quindi Return. Al termine del caricamento potete lanciare il programma con Run. A questo punto sul video comparirà il menù principale mostrato in **figura 1**.

Un esempio

Tanto per farvi un'idea delle possibilità di questo programma, collegate il joystick in porta 2 e selezionate l'opzione Storage del menù principale, spostando opportunamente la leva del joystick

Pagina mancante (pubblicità)

Pagina mancante (pubblicità)

Pagina mancante (pubblicità)

LA PIÙ PREZIOSA COLLEZIONE DI ELETTRONICA COMUNICAZIONE INFORMATICA



11 numeri L. 70.000



9 numeri L. 90.000



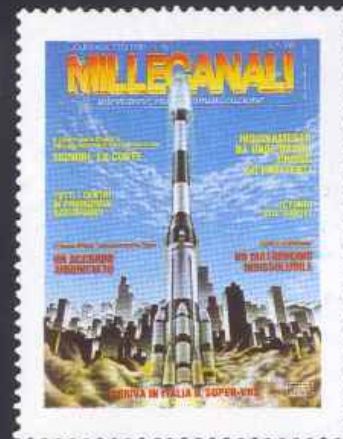
11 numeri L. 60.000



13 numeri L. 75.000



10 numeri L. 56.000



11 numeri L. 65.000



6 numeri L. 29.000



10 numeri L. 59.000



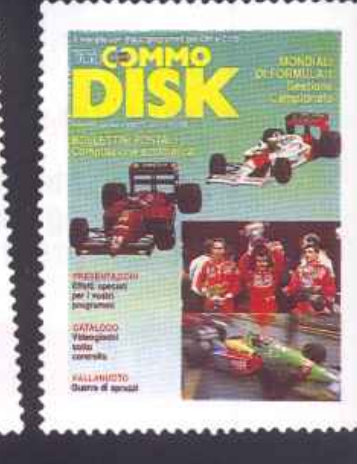
11 numeri L. 55.000



10 numeri L. 120.000



10 numeri L. 145.000



10 numeri L. 125.000



10 numeri L. 54.000



6 numeri L. 64.000



10 numeri L. 150.000

ABBONATI!

Gruppo Editoriale
JCE

Via Ferri, 6
20092 CINISELLO
BALSAMO (MI)

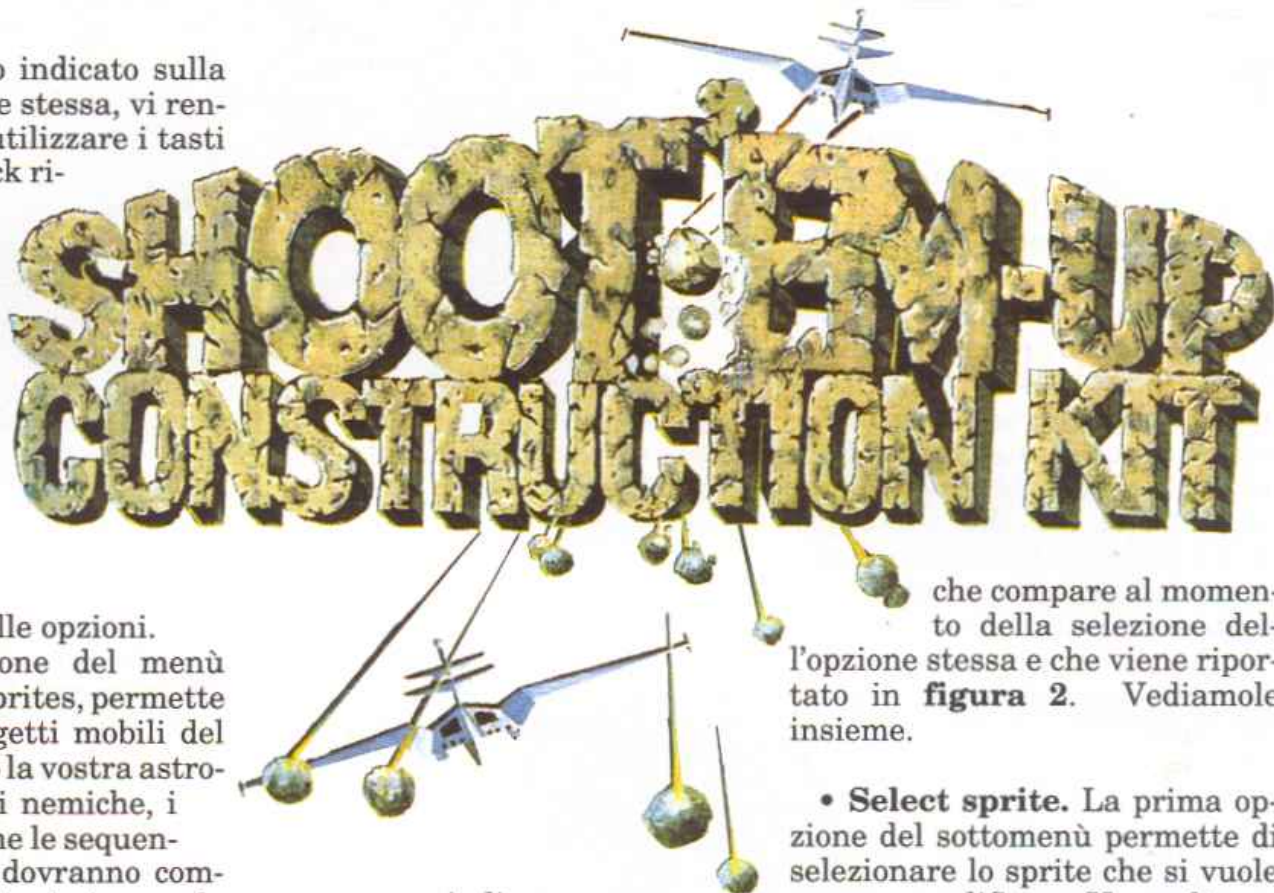
preme il tasto indicato sulla destra dell'opzione stessa, vi renderete conto che utilizzare i tasti al posto del joystick risulta più agevole e soprattutto si risparmia molto tempo.

Quindi il primo consiglio che si può dare è di cercare di sfruttare il più possibile i tasti nella selezione delle opzioni.

La prima opzione del menù principale, Edit Sprites, permette di definire gli oggetti mobili del gioco (per esempio la vostra astronave, le astronavi nemiche, i missili ecc.) e anche le sequenze di oggetti che dovranno comporre le animazioni (come le esplosioni).

Ogni animazione è infatti il risultato di una sovrapposizione di più sprites disegnati in modo tale da risultare, in successione, un unico oggetto animato. Questo è lo stesso principio che sta alla base della realizzazione di film o cartoni animati.

Ogni pellicola infatti è costitui-



che compare al momento della selezione dell'opzione stessa e che viene riportato in **figura 2**. Vediamole insieme.

gramma, quindi se non lo avete ancora caricato fate lo ora, per poter esaminare insieme le varie opzioni del menù principale.

Come editare gli sprite

L'opzione Edit Sprites del menù principale vi permette di editare, cioè di creare o modificare, gli

• **Select sprite.** La prima opzione del sottomenù permette di selezionare lo sprite che si vuole creare o modificare. Una volta selezionata, sulla destra del video compare una griglia di 12x21 caselle che rappresenta la matrice di definizione dello sprite. Tutti gli sprite definibili sono multicolor. Sulla sinistra in alto viene riportato lo sprite stesso così come apparirà nel videogame. Sotto l'immagine dello sprite sono indicati i colori utilizzabili per la definizione dello sprite stesso. I colori possono essere modificati liberamente, spiegheremo tra poco in che modo, tenendo presente i seguenti punti: il colore Background (di fondo) viene utilizzato unicamente per cancellare e non ha alcuna relazione con il colore di fondo dello scenario su cui lo sprite si muoverà.

In altre parole tutto ciò che nella definizione dello sprite appare nel colore Background sarà trasparente; i due colori denominati General sono comuni a tutti gli sprite. Fate quindi molta attenzione alla loro scelta; l'ultimo colore, Changeable, può essere invece scelto liberamente per ciascuno sprite indipendentemente dagli altri.

La serie di colori che compare nella parte bassa dello schermo rappresenta la tavolozza da uti-

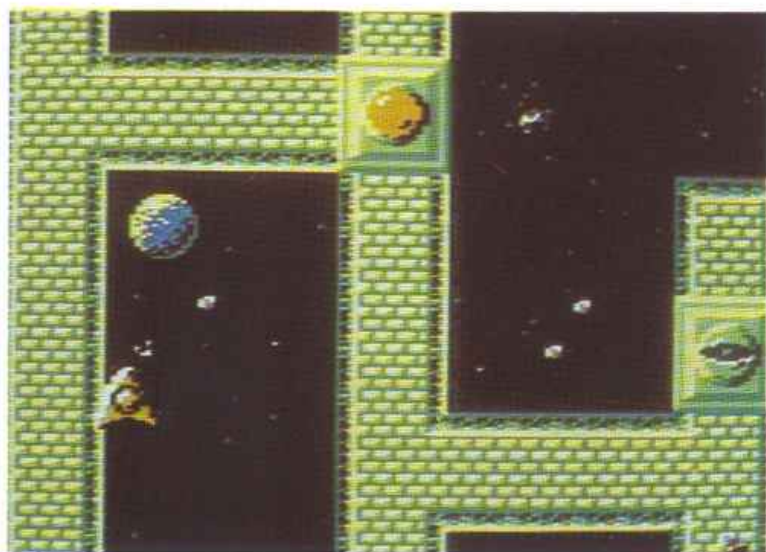


Figura 2.
Il menù
dell'opzione
Edit Sprites.

ta da una enorme sequenza di immagini, i fotogrammi, che viste in rapida successione riescono a riprodurre l'effetto di movimento degli oggetti. Sfrutteremo il videogame in memoria per capire meglio il funzionamento del pro-

sprite presenti in memoria. Potete definire fino a un massimo di 127 sprite, ognuno dei quali risulta individuato da un numero compreso fra 0 e 126. Più precisamente le operazioni che potete compiere sono mostrate nel sotto menù

lizzare per la definizione dei colori descritti. Per selezionare lo sprite da definire o modificare dovete utilizzare il joystick spostando la leva in avanti o indietro per muovervi, rispettivamente, in avanti e indietro nella serie di sprite. Vi consigliamo di effettuare sempre delle prove delle operazioni via via descritte perché è molto importante acquisire dimestichezza con il programma. Solo in questo modo riuscirete a creare velocemente ciò che avete già chiaro in mente. Inoltre ricordatevi sempre che non potrete commettere involontariamente ope-



razioni distruttive; tutte le opzioni pericolose richiedono infatti la conferma prima di operare. Una volta selezionato lo sprite da editare potete fare ritorno al menù principale premendo la barra spazio, e lo stesso vale anche per le opzioni che descriveremo sia di questo che dei prossimi menù.

- **Edit sprite.** La seconda opzione vi permette di agire sullo sprite selezionato. Una volta selezionata l'opzione verrà nuovamente mostrata la schermata descritta in precedenza e sulla griglia di definizione dello sprite comparirà un cursore rettangolare lampeggiante. Il cursore può essere facilmente mosso per tutta la griglia mediante il joystick. Per disegnare nella cella sotto il cursore dovete premere il tasto di Fire.

- **Select colour.** Dovete selezionare questa opzione per scegliere il colore con cui disegnare. Anche in questo caso dopo la selezione si entra nella schermata di lavoro e a questo punto non vi resta che selezionare il colore con cui volete scrivere sulla griglia.

- **Edit colour.** Il colore selezionato può essere modificato mediante questa opzione: una volta rientrati nella schermata di lavoro non dovete fare altro che muovere il cursore sul colore desiderato. L'immagine dello sprite verrà costantemente aggiornata in tempo reale tenendo conto delle vostre scelte quindi in ogni momento ciò che vedrete in alto a sinistra sullo schermo corrisponderà esattamente a quello che vedrete in azione nel videogame.

- **Slide sprite e Mirror sprite.** Queste due

opzioni permettono rispettivamente di far scorrere e riflettere l'immagine dello sprite in una direzione qualsiasi. Una volta attivata una delle due opzioni dovete servirvi del joystick per stabilire la direzione dello scorrimento e della riflessione.

- **Copy sprite.** Per copiare uno dei 127 sprite da una posizione a un'altra della memoria. Questa opzione si rivela particolarmente utile quando si devono sviluppare delle animazioni; in questa situazione infatti è molto utile avere a disposizione più immagini dello stesso sprite in memoria per poter sviluppare più agevolmente una sequenza di animazione. La scelta dello sprite avviene nello stesso modo in cui si seleziona lo sprite da editare. Per confermare la scelta dovete premere il tasto di Fire

e con questa stessa sequenza di azioni potete impostare la posizione di arrivo.

- **Erase sprite.** Con l'ultima opzione si può rimuovere uno sprite dalla memoria. Questa opzione agisce sullo sprite selezionato, cioè sullo sprite che si vede sullo schermo quando si entra nel modo Edit. Data la pericolosità dell'operazione, viene richiesta una conferma prima di procedere alla rimozione. Il menù che compare all'attivazione dell'opzione consente proprio di effettuare questa operazione: selezionando Yes si conferma la scelta mentre selezionando No si fa ritorno al modo Edit Sprites. Per le primissime applicazioni è consigliabile seguire scrupolosamente le operazioni descritte per creare gli sprite. In seguito, dopo aver raggiunto una certa dimestichezza, converrà tener presente che non è necessario ogni volta tornare al menù per selezionare un'opzione ma si può comodamente stare nello schermo di lavoro e effettuare la selezione premendo il tasto corrispondente all'opzione stessa.

Inoltre col joystick è possibile effettuare molto semplicemente l'impostazione dei colori; infatti spostando il cursore verso sinistra si entra nel modo Select colour e, scelto il colore, si può tornare al modo Edit premendo il tasto di Fire oppure spostando la leva del joystick verso destra.

Una volta selezionato il colore si può entrare nel modo Edit colour spostando il cursore in basso.

Provate a verificare quanto è stato detto modificando gli sprite in memoria. Potreste ad esempio cambiare i colori degli sprite oppure aggiungere qualche nuovo elemento cancellandone altri. Se effettuate delle modifiche sugli sprite non dimenticatevi alla fine di registrarle mediante l'opzione Storage (e quindi selezionando la voce Sprites) perché nelle puntate successive gli sprite saranno ancora di scena.

Paolo Gussoni
(continua)



Nasce per facilitare al massimo la gestione degli sprite e dell'alta risoluzione, ma è ricca di molte altre istruzioni per la gestione dei programmi: Sprite&Graphic Basic è un'espansione di grande versatilità.

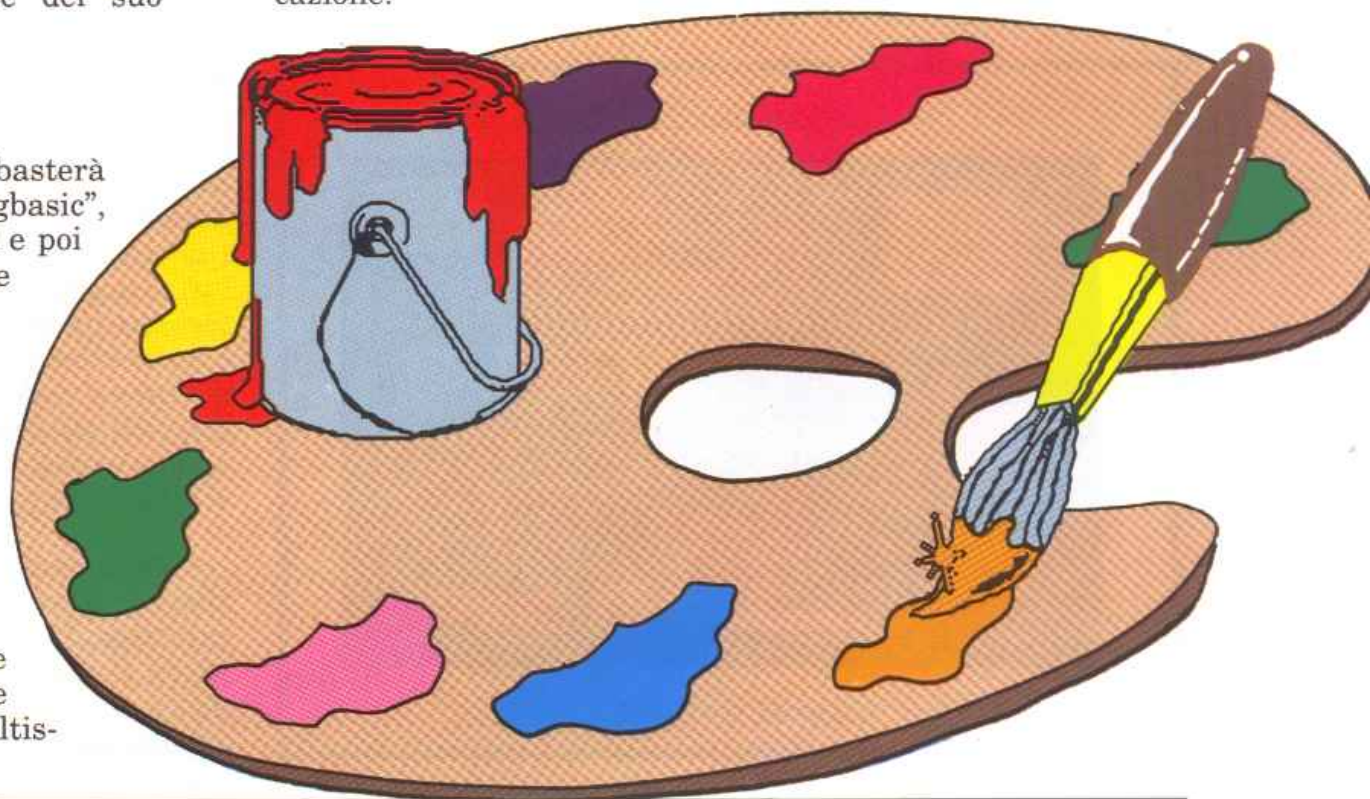
SG Basic: non solo grafica

Sprite&Graphic Basic è una potentissima espansione multipurpose dalle caratteristiche molto particolari. Il nome inganna sulle reali possibilità dell'espansione che non può e non deve essere considerata un semplice tool per la gestione della grafica. Sprite&Graphic Basic è in realtà un vero e proprio ambiente di sviluppo che mette a disposizione dell'utente una serie di potentissimi strumenti per facilitare ogni fase del suo lavoro.

sime istruzioni per facilitare il lavoro di stesura e di debug dei programmi nonché una serie di utilissime istruzioni di controllo per favorire la cosiddetta programmazione strutturata. Dato il numero piuttosto elevato delle istruzioni, la spiegazione della loro funzione verrà fornita in due puntate. In questa come nella puntata successiva le istruzioni saranno raccolte in gruppi secondo l'ambito di applicazione.

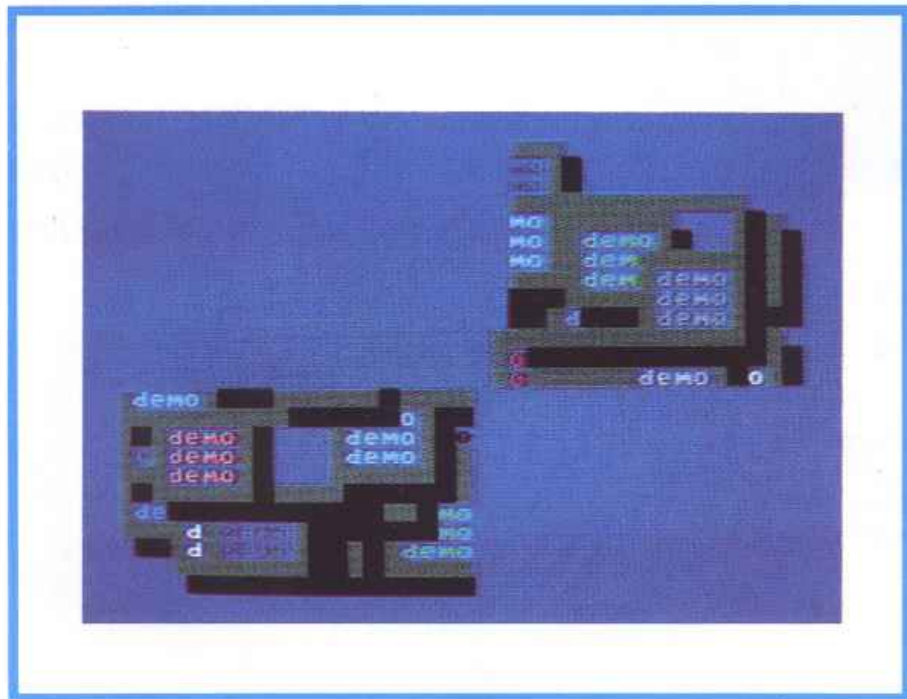
Come funziona

Per attivare SGBasic basterà quindi digitare `Load"sgbasic"`, premere il tasto Return e poi dare il consueto Run. Le istruzioni messe a disposizione da SGBasic sono in tutto cento e permettono di facilitare il lavoro di programmazione in diverse direzioni. Pur essendo stata concepita prevalentemente per facilitare la gestione dell'alta risoluzione e degli sprite, l'espansione mette a disposizione moltis-



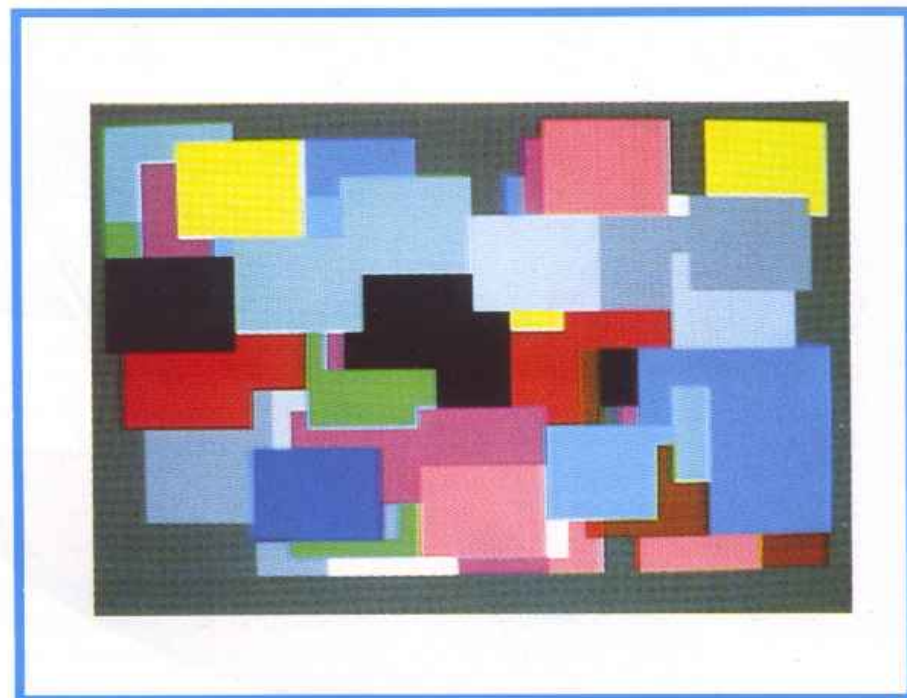
Bassa risoluzione

• **At:** permette di posizionare il cursore in una zona qualsiasi dello schermo e di stampare una stringa. La sintassi è: At x, y, stringa dove x e y rappresentano le coordinate del cursore, mentre stringa è la stringa da visualizzare a partire dalla posizione specificata.



I parametri x e y devono essere interi e compresi rispettivamente nei range 0-39 e 0-24.

• **Set:** permette di visualizzare un carattere in un punto qualsiasi dello schermo e in un determinato colore. La sintassi è: Set cp, cl, x, y dove i primi due para-



metri rappresentano rispettivamente il codice poke del carattere da visualizzare e il colore di visualizzazione dello stesso, mentre gli ultimi due sono le coordinate della posizione in cui deve essere visualizzato il carattere. Per questi due parametri valgono le stesse limitazioni viste per i due parametri omonimi dell'istruzione precedente, mentre cl deve essere un intero fra 0 e 15.

• **Fset:** permette di fissare il colore di linea di una particolare locazione video. La sintassi è: Fset cl, x, y e i parametri hanno lo stesso significato visto per l'istruzione Set.

• **Zset:** permette di fissare il contenuto di una qualsiasi locazione video. La sintassi è: Zset cp, x, y e i parametri hanno lo stesso significato degli omonimi visti nell'istruzione Set.

• **Colour:** permette di fissare il colore del bordo e del fondo dello schermo. La sintassi è: Colour b, f dove b e f rappresentano il codice del colore del bordo e del fondo dello schermo. Entrambi i parametri devono essere compresi fra 0 e 15.

• **Cursor:** permette di fissare il colore di linea, o colore del cursore. La sintassi è: Cursor cl dove cl è il codice del colore di linea.

• **Fill:** permette di colorare una porzione rettangolare del video con un determinato carattere in un determinato colore.

La sintassi è: Fill x1, y1, x2, y2, cp, cl. I primi quattro parametri individuano la porzione rettangolare dello schermo da riempire e rappresentano le coordinate dell'angolo superiore sinistro e inferiore destro di tale porzione. Gli ultimi due parametri rappresentano invece, come nell'istruzione Set, il codice video del carattere con cui riempire la porzione indicata e il suo colore. I primi quattro parametri dovranno rispettare ovviamente i limiti già visti per i parametri simili che compaiono nell'istruzione Set (x1, x2 compresi fra 0 e 39 e y1 e y2 compresi fra 0 e 24).

• **Zfill:** permette di riempire una porzione dello schermo con un determinato carattere. La sintassi è: Zfill x1, y1, x2, y2, cp. Il significato dei parametri è identico a quello degli omonimi parametri dell'istruzione precedente.

- **Ffill**: permette di fissare il colore di linea per una determinata porzione del video. La sintassi è: `Ffill x1, y1, x2, y2, cl`. Il significato dei parametri corrisponde esattamente a quello dei parametri omonimi dell'istruzione `Fill`.

- **Inv**: permette di invertire, cioè di porre in reverse, il contenuto di una porzione dello schermo. La sintassi è: `Inv x1, y1, x2, y2`. I quattro parametri ovviamente definiscono la porzione dello schermo su cui agirà l'istruzione.

- **Up**: permette di effettuare lo scrolling verso l'alto di una determinata porzione dello schermo. La sintassi è: `Up x1, y1, x2, y2` dove i quattro parametri rappresentano la porzione di schermo da scrolare verso l'alto. Occorre tener presente che lo scrolling avviene senza wrap round.

- **Down**: permette di effettuare lo scrolling verso il basso di una determinata porzione dello schermo.

La sintassi è: `Down x1, y1, x2, y2`. Il significato dei parametri è identico a quello degli omonimi parametri dell'istruzione precedente.

- **Left**: permette di effettuare lo scrolling di una determinata porzione dello schermo verso sinistra. La sintassi è identica a quella delle due istruzioni precedenti.

- **Rigth**: permette di effettuare lo scrolling di una determinata porzione dello schermo verso destra. La sintassi è identica a quella vista per le altre istruzioni di scrolling viste in precedenza.

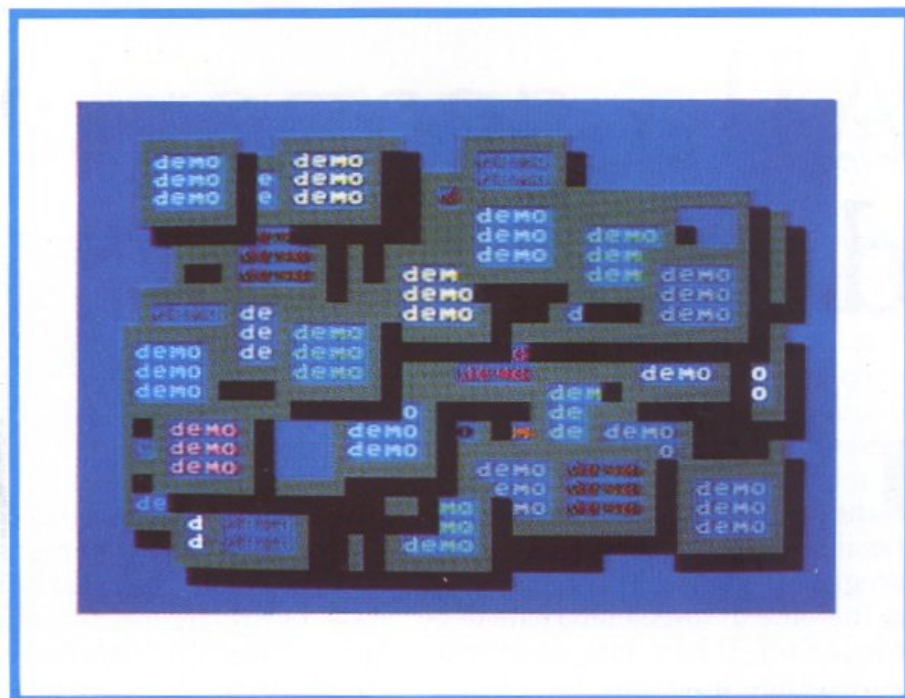
- **Vec**: questa istruzione permette il funzionamento di tutte le istruzioni che operano da interrupt; esse saranno spiegate di seguito. L'istruzione `Vec` non necessita di alcun parametro.

- **Oldvec**: disattiva il meccanismo reso attivo dall'istruzione precedente. Non necessita di alcun parametro.

- **Bflash**: permette di attivare l'effetto flashing del bordo dello schermo. L'istruzione agisce da interrupt e quindi, una volta eseguita, il bordo comincerà a lampeggiare automaticamente sia in modo programma sia in modo diretto. La sin-

tassi è: `Bflash cl1, cl2, tm`. I primi due parametri specificano i codici dei colori che dovranno alternarsi nell'effetto flashing, mentre `tm` rappresenta la frequenza della pulsazione in decimi di secondo.

- **Gflash**: è analoga all'istruzione precedente, nel senso che permette di attiva-



re l'effetto flashing del fondo dello schermo. La sintassi è quindi identica a quella dell'istruzione precedente.

- **Bgflash**: attiva l'effetto flashing contemporaneamente per il bordo e per il fondo dello schermo. La sintassi è identica a quella delle due istruzioni precedenti.

- **Foff**: disattiva l'effetto flashing. Questa istruzione disattiva l'effetto delle tre istruzioni precedenti senza invece avere alcun effetto sulle altre istruzioni che agiscono sul meccanismo di interrupt.

- **Flash**: permette di attivare l'effetto flashing per tutti i caratteri visualizzati in un determinato colore. La sintassi è: `Flash cl1, cl2, cl3`. Dopo l'esecuzione dell'istruzione tutti i caratteri visualizzati in uno dei tre colori specificati con i parametri cominceranno a lampeggiare. La frequenza della pulsazione è fissa.

- **Off**: disattiva l'effetto dell'istruzione precedente.

Daniele Maggio
(continua)

Nella scorsa puntata si è esaminato il microprocessore e i principali segnali presenti nel calcolatore; si è anche visto come il microprocessore legge una a una le istruzioni costituenti lo stesso programma e come le esegua. Vale ora la pena di ritornare sull'argomento e di approfondirlo.

Alla scoperta dei bus

Innanzitutto occorre sapere che il microprocessore comunica con gli altri dispositivi attraverso un insieme di linee elettriche detto bus dati. Il bus dati, costituito da otto linee, è costantemente percorso da informazioni, sotto forma di impulsi elettrici, in entrambe le direzioni. Ciò significa che ci saranno dei dati che scritti dal microprocessore verranno letti da un dispositivo esterno, mentre altri dati saranno scritti da qualche interfaccia o memoria e verranno letti dal microprocessore. In que-

sto senso si dice che il bus dati è bidirezionale ovvero percorso dalle informazioni in entrambi i sensi.

La bidirezionalità non implica quindi che gli impulsi elettrici possano avere diverse direzioni di propagazione, ma semplicemente che il bus potrà vedere i diversi dispositivi che lo pilotano ora come uscite, ora come entrate, a seconda della direzione del flusso dell'informazione. Poiché il bus dati è costituito da otto linee elettriche, ognuna delle quali potrà essere in due stati (stato alto corrisponden-

te a una tensione maggiore di 2,5 volts e stato basso con tensione inferiore a 0,8 volts, che corrispondono rispettivamente agli stati binari 1 e 0), esso si potrà trovare in $256 (2^8)$ stati diversi, dove ogni stato differisca dagli altri per almeno il valore di una linea. In ogni istante sarà possibile abbinare a ogni linea una cifra binaria (0 o 1) mentre l'insieme delle otto linee, costituente il bus, potrà essere abbinato a un numero binario di otto cifre detto byte.

Ogni linea del bus dati sarà distinguibile dalle altre da un apposito numero che indicherà la posizione del bit trasmesso all'interno del byte. In questo senso la linea inferiore del bus dati sarà indicata come D0, e a essa corrisponde-

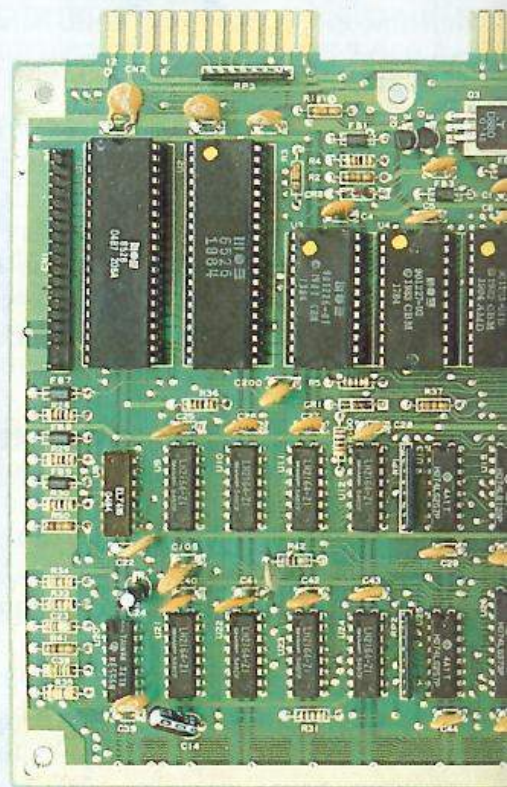
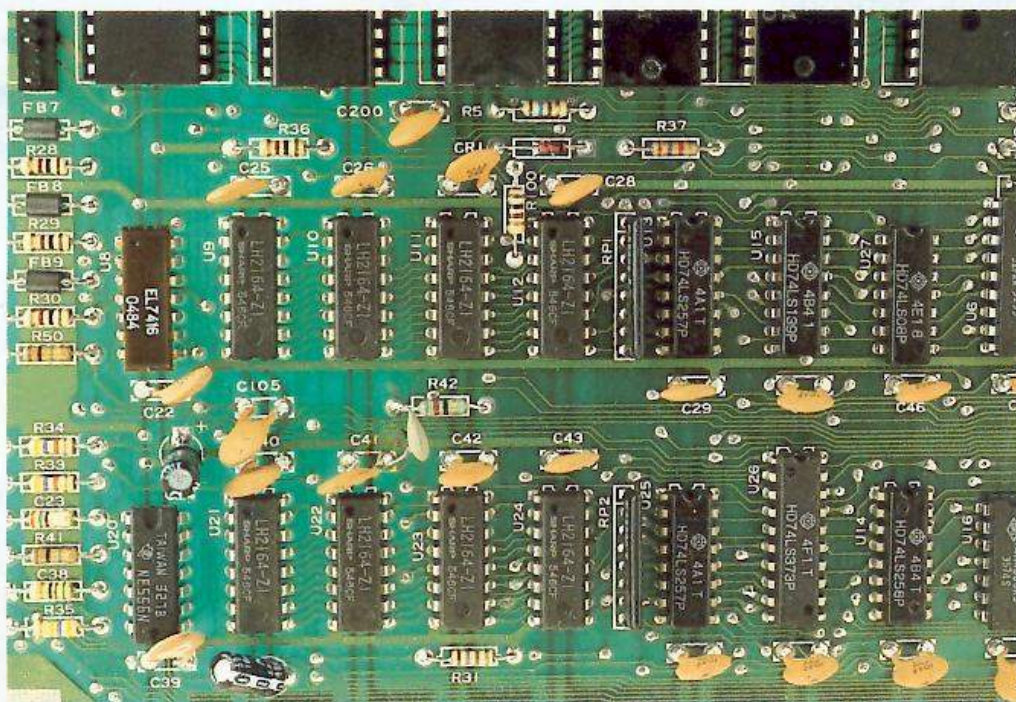


Figura 2a. La memoria ROM del C64.

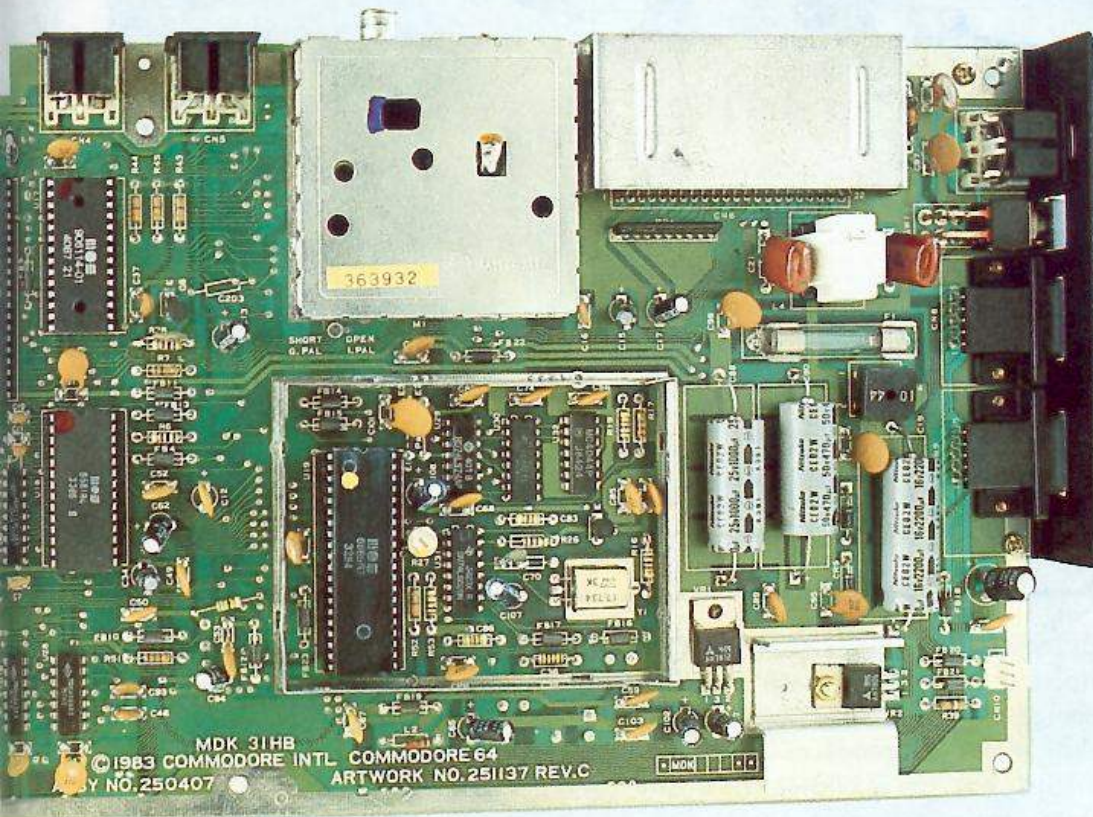


rà nel byte trasmesso il bit di peso minore. Le rimanenti linee avranno nome: D1, D2, D3, eccetera, sino all'ultima linea, che trasporterà il bit più significativo di nome D7. Poiché il bus dati trasporta informazioni nel formato byte, e un byte potrà trovarsi in 256 stati diversi, si potrà abbinare

co al quale intende inviare o ricevere dei dati tramite il bus dati. Il bus indirizzi è costituito da un insieme di 16 linee elettriche che consentono di trasportare un'informazione di 16 bit, ovvero di 2 byte. Poiché lo stato assunto da ognuna delle 16 linee è indipendente da quello delle rimanenti 15, il

bus indirizzi potrà trovarsi in 65536 (2^{16}) stati distinguibili. Se a ognuno di questi stati si associa un numero, risulta che il bus indirizzi è in grado di trasportare un numero compreso tra 0 e 65535. Le linee costituenti il bus indirizzi sono numerate con lo stesso sistema utilizzato per il bus dati. La linea a cui corrisponderà il bit di peso minore sarà indicata come A0, quella successiva come A1, sino ad arrivare ad A15 che sarà la linea abbinata al bit più significativo.

Una caratteristica tipica del bus indirizzi è quella di essere unidirezionale: il bus vedrà quindi il microprocessore sempre come uscita e gli altri dispositivi come entrate, giacché l'informazione dell'indirizzo si propagerà sempre dal microprocessore agli altri dispositivi. Siccome l'indirizzo propagato dal bus indirizzi potrà avere 65536 differenti valori, tale numero sarà anche quello dei dispositivi logici indirizzabili, ognuno dei quali verrà ora indicato con il più appropriato termine di registro. I 65536 registri coi quali il microprocessore potrà comunicare saranno ripartiti fra diversi dispositivi fisici (circuiti integrati) presenti nel C64, ragion per cui il bus indirizzi, a differenza del bus dati, non verrà interamente inviato a ogni dispositivo fisico ma



re a ogni possibile stato del byte un numero compreso tra 0 e 255 che indicherà in modo univoco l'informazione trasmessa. Il bus dati del microprocessore va a connettersi direttamente con la maggior parte dei dispositivi presenti nel calcolatore, per l'esattezza con tutte le ROM, le RAM e le interfacce, assicurando così la comunicazione fra questi dispositivi e il microprocessore stesso.

Il bus indirizzi

Poiché moltissimi dispositivi fanno capo allo stesso bus dati e poiché, nel corso dell'elaborazione di un programma, il microprocessore necessita di comunicare con un solo dispositivo per volta, oc-

correrà un altro bus, detto bus indirizzi, che permetta al microprocessore di erogare l'informazione concernente il dispositivo speci-

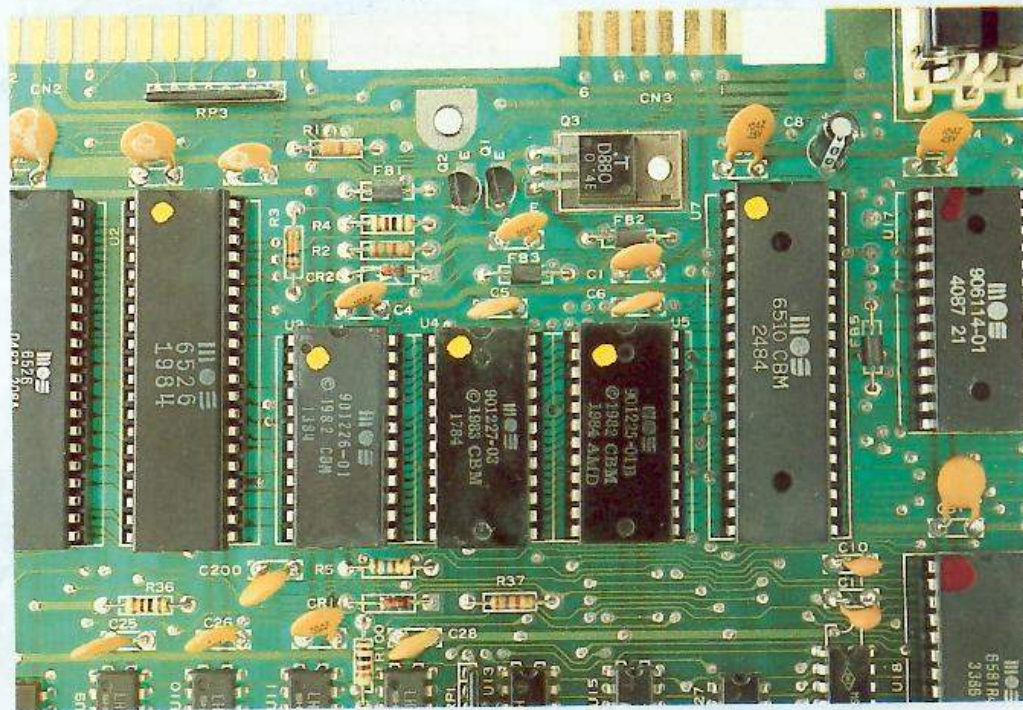


Figura 1a. Le tre ROM presenti nel C64.

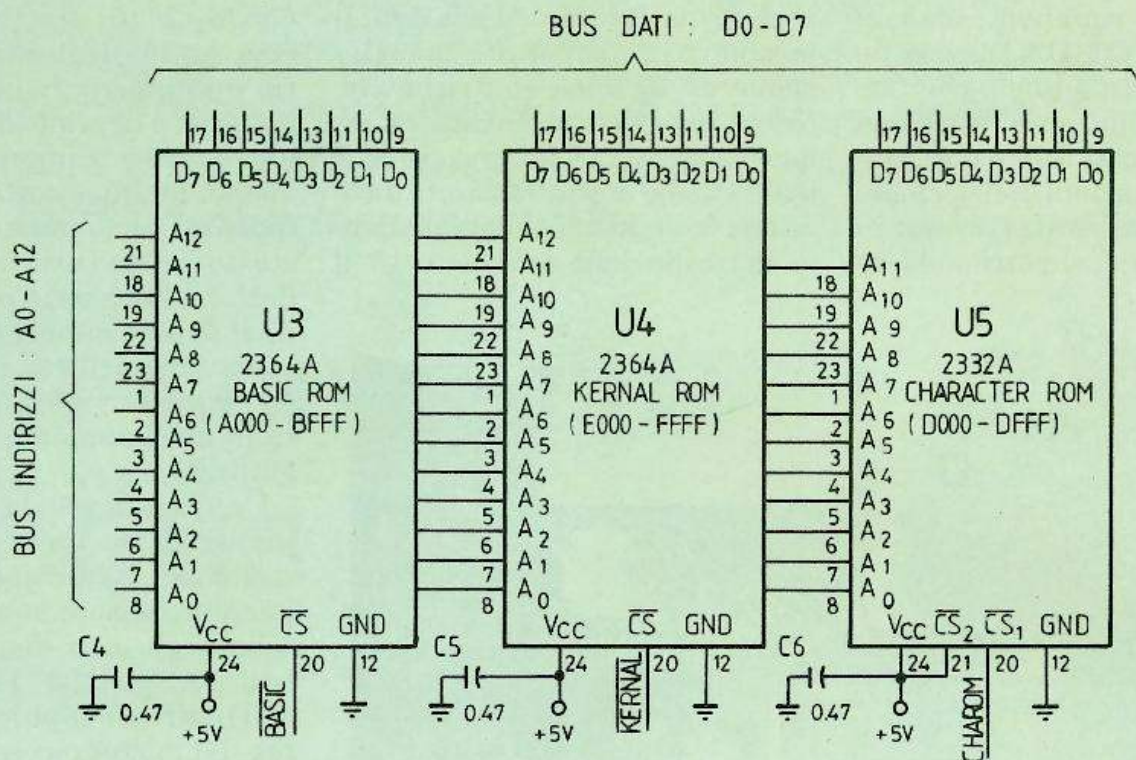


Figura 1b. Ecco tutti i collegamenti delle tre ROM.

dovrà subire una prima manipolazione da parte della logica d'indirizzamento implementata quasi totalmente nel PLA (Programmable Logic Array) MOS 906114. In questa puntata si esamineranno le ROM e le RAM, mentre le interfacce saranno esaminate nella prossima puntata.

Le ROM

La figura 1a rappresenta le tre ROM presenti nel C64, mentre la figura 1b illustra tutti i collegamenti di questi integrati. Innanzitutto occorre ricordare che le ROM sono memorie di sola lettura, quindi il contenuto dei registri presenti in esse può venir letto solo dal microprocessore, essendo stato scritto una volta per tutte all'atto di fabbricazione dello stesso integrato. Le tre ROM contengono sia codice eseguibile, sia dati. Per l'esattezza la prima ROM (MOS 901226) contiene il codice dell'interprete Basic, il cui scopo è di tradurre e di eseguire tutte le istruzioni di un programma in quel linguaggio. La seconda ROM

(MOS 901227) detta Kernal ROM implementa invece tutto il codice necessario alla gestione della RAM e dei dispositivi di Input/Output, ovvero il codice necessario a pilotare il registratore, la scansione della tastiera, la visualizzazione su monitor, e la comunicazione attraverso sia il bus seriale sia la porta parallela. La terza ROM (MOS 901225), detta Character ROM, contiene i dati necessari alla visualizzazione sullo schermo dei vari caratteri. Le prime due ROM, essendo delle 2364, conterranno al loro interno 64 Kilobit articolati in 8192 (2^{13}) registri di 8 bit ciascuno. Tali ROM sono quindi della capacità di 8 Kb.

A esse, oltre alle linee di alimentazione, faranno capo i seguenti segnali:

- D0-D7: il bus dati, le cui otto linee saranno connesse internamente a tutti gli 8192 registri.
- A0-A12: le 13 linee inferiori del bus indirizzi che permetteranno di selezionare uno fra gli 8192 registri presenti. CS che indicherà, abbassandosi, l'avvenuto selezio-

namento della ROM; e la terza ROM, quella dei caratteri, del tipo 2332, che conterrà invece solo 4096 registri e quindi avrà una capacità di 4 Kb.

A essa faranno capo le stesse linee che raggiungevano le due ROM precedenti, con la differenza che il bus indirizzi avrà ora una linea in meno (12 da A0 ad A11) essendosi ridotto alla metà il numero dei registri selezionabili. Sia il bus dati sia la parte utilizzata del bus indirizzi sono comuni a tutte e tre le ROM e le collegano al microprocessore, affinché questi possa indirizzarle e leggerle.

I tre \sim CS, separati e distinti faranno invece capo a tre terminali di uscita del PLA, il quale abbassando uno di questi segnali, selezionerà quella fra le tre ROM che il microprocessore intende leggere. Il funzionamento delle tre ROM è abbastanza semplice: in un primo tempo esse riceveranno dal microprocessore la parte inferiore del bus indirizzi che, decodificato all'interno della ROM stessa, selezionerà un particolare registro fra tutti quelli presenti;

successivamente, all'abbassarsi del $\sim CS$, che come dice la stessa sigla selezionerà il particolare integrato con cui il microprocessore intende comunicare, le tre ROM invieranno sul bus dati il contenuto di tale registro precedentemente selezionato.

Quale dei $\sim CS$ debba venir selezionato sarà deciso dalla logica d'indirizzamento presente all'interno del PLA in funzione della parte alta del bus indirizzi e dei valori assunti da altre linee di controllo, come avremo modo di vedere più avanti quando ci occuperemo del PLA in particolare.

Le RAM e i MUX

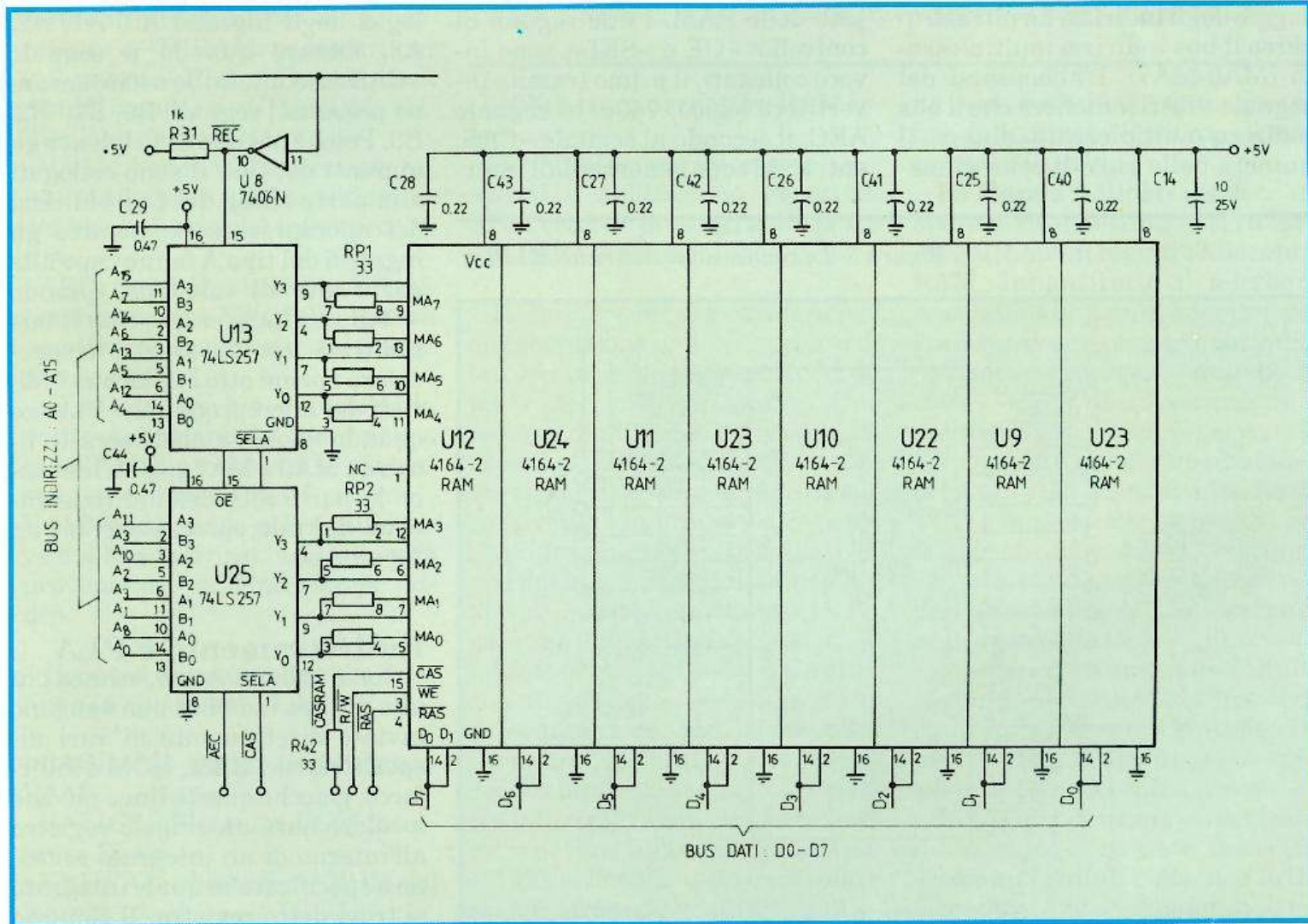
Le RAM, lo ricordiamo per chi non lo sapesse, sono delle memorie di lettura e scrittura: questo significa che il microprocessore può

scrivere il contenuto dei diversi registri per poi rileggerlo in un secondo tempo. L'operazione di lettura non comporta la perdita dell'informazione presente nel registro, che ivi rimane sino a che non venga effettuata un'operazione di scrittura in detto registro o non si spenga il calcolatore. La memoria RAM è rappresentata nella **figura 2a**, mentre lo schema elettrico è illustrato nella **figura 2b**. Il banco RAM montato nel C64 è costituito da otto RAM 4164 (nei modelli più recenti queste sono state sostituite con due RAM 41464). Le RAM 4164 sono memorie da 64 Kb: ognuno di questi integrati conterrà quindi 65536 celle di memoria che potranno essere individualmente lette e scritte. A differenza delle ROM presenti nel C64 queste RAM non sono organizzate internamente nel formato byte,

ragion per cui a ognuna di esse farà capo una sola linea del bus dati. In questo senso la prima RAM (u21) conterrà tutti bit di posizione 0 (facenti capo a D0), di tutti i registri costituenti la RAM. Sempre per la stessa ragione, un registro ubicato in RAM vedrà le otto celle di memoria che lo compongono distribuite negli otto integrati che costituiscono la RAM. Quindi la lettura, o scrittura, di un registro ubicato in RAM comporterà la selezione di tutti gli otto integrati, ognuno dei quali fornirà un singolo bit del byte totale costituente l'informazione presente in quel registro. Poiché in ogni integrato costituente la RAM sono memorizzati 65536 bit, sarebbe stato logico aspettarsi un bus indirizzi di ben 16 linee atto a selezionare ognuna di queste celle.

Tuttavia poiché un bus di tale

Figura 2b. Lo schema elettrico della memoria RAM.



dimensioni avrebbe sovradimensionato il numero dei piedini presenti sull'integrato, si è pensato di ridurre tale bus a sole otto linee (nello schema da MA0 a MA7) con l'avvertenza di fornire l'intero indirizzo di 16 bit in due tempi. Questo bus indirizzi ridotto è detto multiplessato e fa capo a tutti e otto gli integrati costituenti la RAM. Altri segnali presenti sugli integrati della RAM sono: D1, D0, R/W, ~RAS e ~CAS. I piedini D1 e D0 fanno capo rispettivamente alle entrate e alle uscite delle diverse celle di memoria, e vanno a collegarsi, esternamente all'integrato, sulla linea del bus dati che serve quel particolare integrato RAM. Il segnale R/W, proveniente dal microprocessore, indicherà invece alla RAM se si intende effettuare una operazione di lettura o di scrittura: se tale segnale sarà alto l'operazione sarà lettura, viceversa si tratterà di un accesso in scrittura. Infine i segnali ~RAS e ~CAS arbitreranno il multiplessaggio degli indirizzi forniti attraverso il bus indirizzi multiplessato (MA0-MA7). L'abbassarsi del segnale ~RAS indicherà che il bus indirizzi multiplessato conterrà il numero della riga di celle di me-

moria a cui si intende accedere, mentre il passaggio allo stato basso di ~CAS sanzionerà la presenza del numero della colonna in cui si trova la cella selezionata sul bus indirizzi multiplessato, e attiverà l'operazione di lettura o scrittura del dato stesso, a seconda dello stato assunto dalla linea R/W. Come abbiamo visto precedentemente, il bus indirizzi in uscita al microprocessore non è multiplessato, ragion per cui non potrà essere collegato direttamente al banco RAM. Il compito di multiplessare il bus indirizzi del microprocessore è affidato a due integrati multiplexer (MUX: 74LS257 u13 e u25 nello schema). Ognuno di questi due integrati presenta otto entrate, quattro uscite e due terminali di controllo. Le entrate sono indicate con A0, B0, A1, B1, A2, B2, A3, B3 e vanno a collegarsi al bus indirizzi del microprocessore, mentre le uscite, indicate con le lettere Y0, Y1, Y2, Y3, fanno capo al bus indirizzi multiplessato delle RAM. I due segnali di controllo: ~OE e ~SELA sono invece collegati, il primo tramite invertitore logico (7406) al segnale AEC, il secondo al segnale ~CAS; entrambi sono generati dall'inter-

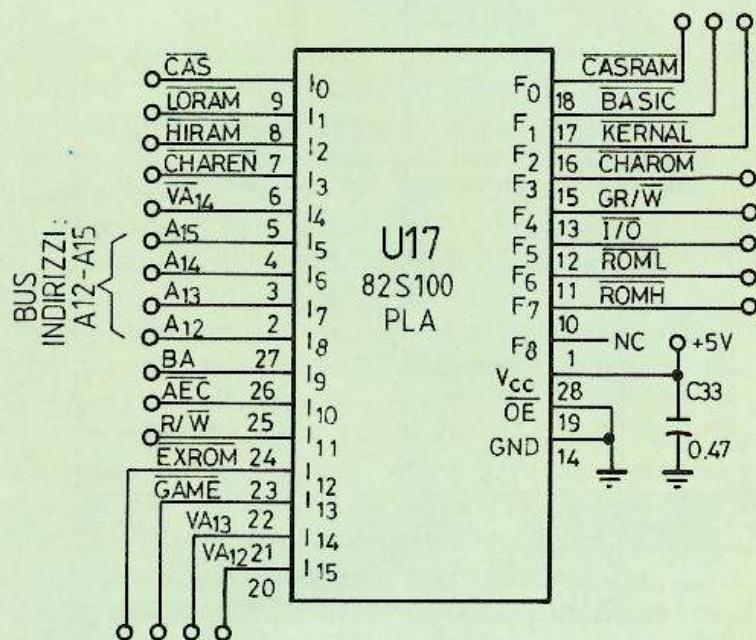
faccia video. Il funzionamento di questi integrati è piuttosto semplice: innanzitutto se il segnale presente su ~OE è alto le uscite di questo integrato saranno nello stato ad alta impedenza, ragion per cui il microprocessore non potrà indirizzare il banco RAM.

Questo succederà sul fronte basso di AEC che, come avevamo visto nella scorsa puntata, commuta l'uso dei bus fra il microprocessore e l'interfaccia video; ciò si verificherà perché durante il periodo in cui ~AEC rimarrà a livello basso l'interfaccia video dovrà indirizzare il banco RAM, e quindi i multiplexer che supportano l'indirizzamento da parte del microprocessore dovranno essere scollegati. Il secondo segnale che controlla i MUX, ~SELA, ha il compito di commutare gli otto segnali in ingresso sulle quattro uscite: per l'esattezza, quando ~CAS (che fa capo a ~SELA) sarà basso le quattro uscite Y0, Y1, Y2, Y3 presenteranno gli stessi stati logici degli ingressi A0, A1, A2, A3, mentre quando il segnale ~CAS sarà alto sulle uscite saranno presenti i segnali: B0, B1, B2, B3. Poiché nei due multiplexer gli ingressi del tipo B sono collegati alla parte bassa del bus indirizzi del microprocessore, mentre gli ingressi del tipo A fanno capo alla parte alta di tale bus, quando ~CAS risulterà essere alto il bus indirizzi multiplessato rispecchierà i primi otto bit del bus indirizzi del microprocessore, mentre quando ~CAS si abbasserà le linee da MA0 a MA7 rispecchieranno la parte alta dell'indirizzo inviato dal microprocessore: bit da A8 ad A15.

Indirizzamento e PLA

Come abbiamo visto, le linee più alte del bus indirizzi non vengono inviate direttamente ai vari dispositivi fisici RAM, ROM e interfacce, giacché queste linee più che a selezionare un singolo registro all'interno di un integrato servono a specificare in quale integrato si trovi detto registro. Il compito

Figura 3. Le connessioni elettriche del PLA.



del circuito, detto PLA (MOS 906114), è appunto di decodificare la parte alta dell'indirizzo del bus indirizzi del microprocessore per attivare successivamente la linea \sim CS dell'integrato nel cui interno si trova il registro indirizzato. Le connessioni elettriche del PLA sono illustrate nello schema elettrico in **figura 3**. Il PLA presenta 16 terminali d'ingresso, collegati con le linee più alte del bus indirizzi e con tutta una serie di linee di controllo, e otto terminali di uscita che andranno a collegarsi più o meno direttamente a vari dispositivi da selezionare. Una sola uscita del PLA potrà essere bassa in un certo istante, e quale essa sia dipenderà dallo stato logico di tutte le entrate. Al PLA, come segnali in entrata, fanno capo le seguenti linee elettriche:

\sim CAS : proveniente dal VIC necessario alla genesi del segnale CASRAM da inviare al banco.

RAM \sim LORAM, generati dall'interfaccia presente nel microprocessore e \sim HIRAM: necessari per informare il PLA delle eventuali disabilitazioni o \sim CHAROM: abilitazione delle ROM.

\sim VA14 : banco video selezionato. A12-A15: linee più significative del bus indirizzi del microprocessore.

\sim AEC: commutazione bus fra microprocessore e VIC R/W: lettura o scrittura proveniente dal microprocessore.

\sim EXROM: linee provenienti dalla porta espansioni che comunicano la \sim GAME: presenza e le dimensioni di eventuali ROM esterne.

VA12-VA13: linee più significative del bus indirizzi multiplessato proveniente dall'interfaccia video.

Ecco qui di seguito l'elenco delle linee in uscita dal PLA:

\sim CASRAM: abilitazione banco RAM

\sim BASIC: abilitazione ROM interprete Basic

\sim KERNAL: abilitazione ROM del Kernal

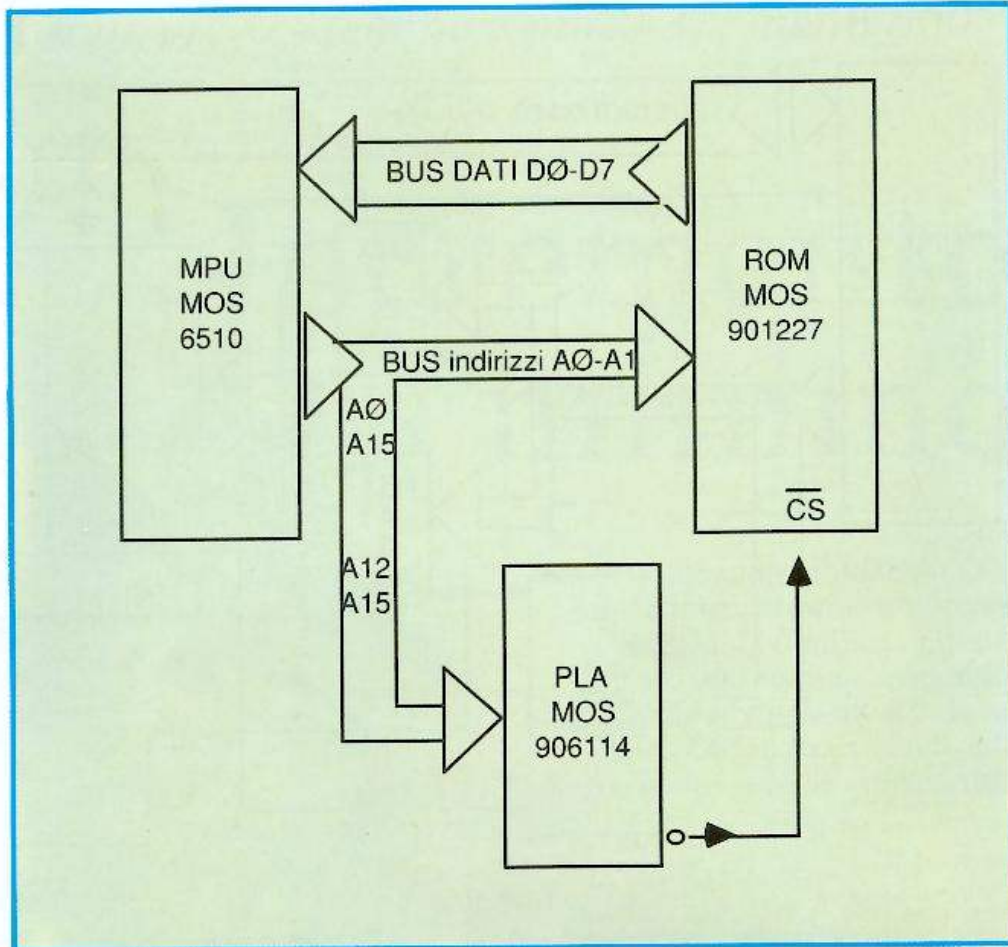


Figura 4. Schema del principio dell'operazione di lettura di una ROM.

GR/W: lettura e scrittura RAM colore

\sim I/O: abilitazione interfacce

\sim ROML: abilitazione eventuali ROM esterne presenti sulla porta

\sim ROMH: espansioni.

Esamineremo ora la sequenza dei segnali in una operazione di lettura in una generica ROM da parte del microprocessore. Uno schema di principio dell'operazione è illustrato nella **figura 4**. Come si può vedere il microprocessore formula l'indirizzo del registro al quale vuole accedere sulle 16 linee del bus indirizzi. Le linee più alte di questo indirizzo raggiungeranno il PLA, il quale in base allo stato di queste linee e delle altre che raggiungono i suoi ingressi determinerà quale dispositivo selezionare abbassando la corrispondente uscita. La ROM selezionata, utilizzando la parte inferiore del bus indirizzi (linee da A0 a A12), selezionerà un particolare registro fra tutti quelli che contie-

ne e ne copierà il contenuto sul bus dati affinché il microprocessore possa leggerlo.

La **figura 5** illustra invece un'operazione di lettura (o scrittura) eseguita su un registro allocato in RAM. Innanzitutto il microprocessore invierà sull'omonimo bus l'indirizzo del registro col quale intende comunicare, quindi nel caso si tratti di un'operazione di scrittura abbasserà il segnale R/W. Durante queste operazioni il segnale \sim AEC sarà tenuto alto dal VIC, e indicherà che l'uso dei bus è lasciato alla MPU. I segnali \sim RAS e \sim CAS, generati anch'essi dall'interfaccia video, saranno nello stato alto.

Poiché \sim CAS sarà alto i multiplexer genereranno sul bus indirizzi multiplessato (MA0-MA7) i primi otto bit dell'indirizzo presente sul bus del microprocessore (A0-A7). Successivamente l'interfaccia video abbasserà il segnale \sim RAS su ognuna delle otto RAM, facendo sì che queste memorizzi-

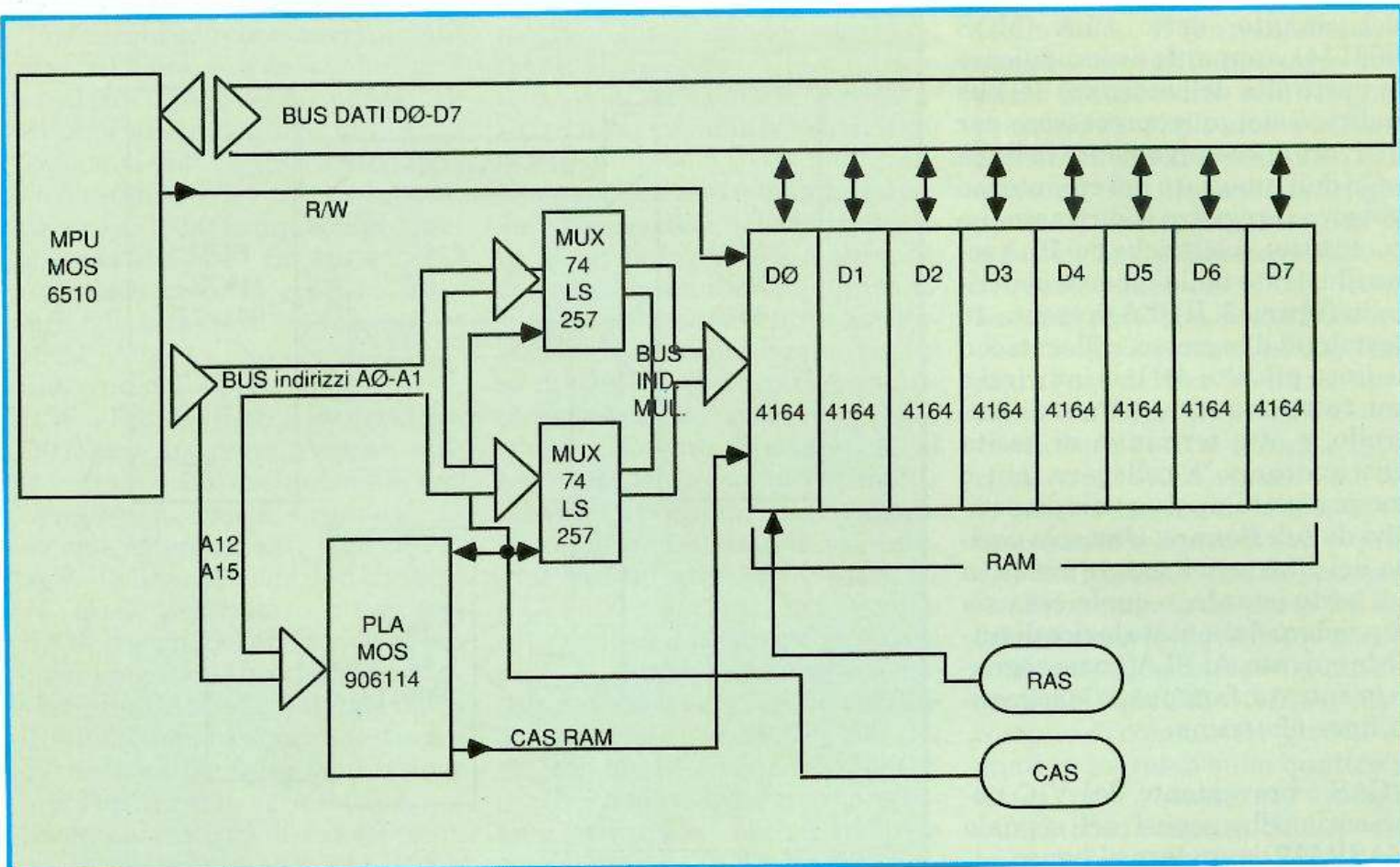


Figura 5. Operazione di lettura o di scrittura eseguita su un registro in RAM.

no l'indirizzo come quello della riga in cui si trova la cella di memoria a cui il microprocessore intende accedere. Quindi il VIC abbasserà anche \sim CAS, con l'effetto di commutare i multiplexer: ora sul bus indirizzi multiplessato saranno presenti gli otto bit più significativi dell'indirizzo (A8-A15). Anche il PLA riceverà \sim CAS e, in base allo stato delle linee alte del bus indirizzi nonché di tutte le altre linee di controllo che a lui giungono come ingressi, deciderà se il registro selezionato si trovi più o meno in RAM. Se è così, abbasserà il segnale \sim CASRAM il quale informerà ognuna delle 8 RAM che l'indirizzo presente sul bus multiplessato è quello della colonna in cui si trova la cella di memoria selezionata. Quando le RAM percepiranno l'abbassamento di \sim CASRAM, se la linea R/W sarà impostata alta (operazione di lettura), invieranno il contenuto della cella di memoria selezionata sulla corrispondente linea del bus

dati. Viceversa, se il segnale R/W risulterà essere basso (operazione di scrittura), lo stato della linea del bus dati facente capo a quella RAM verrà memorizzato nella cella di memoria selezionata. Da quanto detto, emerge che la linea \sim RAMCAS svolge una duplice funzione: innanzitutto informa la RAM che l'indirizzo presente sul bus multiplessato è quello della colonna in cui reperire la cella di memoria selezionata; in secondo luogo svolge la funzione di \sim CS (abilitazione di circuito) determinando l'esecuzione dell'operazione di lettura o scrittura su quel particolare integrato. Infatti, se il registro selezionato non fosse ubicato in RAM, i segnali \sim CAS, \sim RAS e R/W avrebbero sempre lo stesso andamento visto precedentemente, mentre il segnale \sim CASRAM rimarrebbe sempre nello stato alto. Qualcuno, esaminando il proprio C64 potrebbe anche non trovarvi né il PLA 906114, né le RAM 4164, né i MUX 74LS257.

Infatti i modelli di più recente costruzione montano solo due RAM, ognuna delle quali ha una capacità di 256 Kilobit distribuiti in 65536 registri della lunghezza di quattro bit. Ognuna di queste nuove RAM sarà collegata al bus indirizzi multiplessato e alle linee di controllo come le RAM precedenti; la differenza consiste nel fatto che i terminali di ingresso e uscita dati D1 e D0 sono stati sostituiti con quattro terminali bidirezionali D0, D1, D2, D3 che fanno capo direttamente a quattro linee del bus dati. L'assenza dei MUX come circuiti fisici (74LS257) denota che essi sono stati integrati, insieme al resto della logica d'indirizzamento, all'interno di un nuovo PLA.

Sergio Fiorentini
(continua)

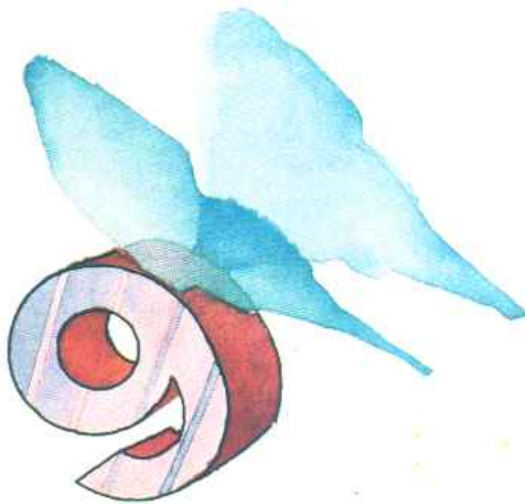
Attenzione: il simbolo \sim va inteso come una barretta sopra la sigla. Per esempio: \sim CAS = CAS.

Lo sapevate che 600 milioni più 0.2, oppure più 0.37, meno 600 milioni fa 0.25? Certo che non lo sapevate: è assurdo. Eppure il vostro Commodore la pensa così...

Incremento minimo

Se qualcuno vi chiedesse di eseguire a memoria questo semplice calcolo: 600 milioni più 0.2 meno 600 milioni, rispondereste ovviamente 0.2. Il vostro C64 invece no. A fronte delle stesse operazioni





stampa 0.25. Anche se gli chiedeste:

```
PRINT 600000000 + 0.37 - 600000000
```

risponderebbe ancora 0.25.

Tutto questo vi deluderebbe perché vi indurrebbe a sospettare che il C64 non sia un buon calcolatore e che alle volte non riesca nemmeno a effettuare correttamente le operazioni più semplici. Il fatto è che tutti i calcolatori incorrono in imprecisioni o arrotondamenti più o meno sensibili, in dipendenza del numero di cifre che possono gestire.

Le imprecisioni o arrotondamenti hanno naturalmente una spiegazione, ma per trovarla bisogna prendere le cose un po' alla lontana. Il manuale d'uso, di cui viene dotato ogni acquirente del C64, fornisce i seguenti dati:

- Il C64 può lavorare con numeri che contengono 10 cifre, ma quando un numero viene stampato, vengono visualizzate solo nove cifre.
- I numeri compresi fra 0.01 e 999999999 (positivi o negativi) sono stampati usando la notazione standard. I numeri al di fuori di questo campo vengono stampati usando la notazione scientifica.
- I limiti entro i quali il C64 può gestire la notazione scientifica sono:

- numero più grande: $\pm 1.70141183E + 38$
- numero più piccolo: $\pm 2.93873588E - 39$

Inoltre il manuale d'uso specifica che la notazione scientifica non è altro che un modo per esprimere un numero molto grande o molto piccolo sotto forma di potenza di 10. Essa si effettua mediante l'utilizzo della lettera E seguita da un numero positivo o negativo e, per convenzione, il numero che precede la E va moltiplicato (se dopo la E c'è il segno +), oppure diviso (se dopo la E c'è il segno -) per 10 elevato alla potenza rappresentata dal numero che segue la E. In pratica per passare dalla notazione scientifica a quella standard:

- se il numero è un intero e la E è seguita dal segno + si aggiunge al numero stesso una quantità di zeri pari al numero che segue la E; se il numero è decimale si sposta la virgola di altrettanti posti verso destra;
- si opera in senso opposto se la E è seguita dal segno -.

Esempi:

```
11E + 07 = 110000000
11.13245E + 04 = 111324.5
11E - 05 = 0.00011 = 1.1E-04
1113245E - 03 = 113.245
```

Tenendo presente che i numeri positivi o negativi da 0.01 a 999999999, benché visualizzati in notazione standard, sono compresi nei limiti indicati nel manuale, per la notazione scientifica (per esempio $114 = 11.4E + 01$) possiamo scrivere che il campo dei numeri utilizzabili dal C64 si divide in due gruppi che si estendono:

da $-1.70141183E+38$
a $-2.93873588E-39$

per i numeri negativi, e

da $+2.93873588E-39$
a $+1.70141183E+38$

per i numeri positivi.

Fra un gruppo e l'altro di numeri sta lo zero, esso cioè si trova fra $-2.93873588E-39$ e $+2.93873588E-39$.

Entro questo intervallo tutti i numeri sono considerati zero dal computer, vale a dire lo zero li rappresenta tutti. Se digitiamo:

```
A = 2.938873588E - 39 (primo numero positivo riconosciuto dal computer)
```

```
B = 2.93873587E - 39 (ultimo numero positivo interno all'intervallo)
```

e quindi:

```
PRINT A*2;B*2
```

avremo come risposte: $5.87747176E-39$ e 0. Ossia, il C64 esegue esattamente l'operazione relativa al primo numero in quanto è in grado di riconoscerne l'effettiva entità mentre sbaglia la seconda operazione perché considera uguale a zero il secondo numero, essendo lo stesso compreso nel predetto intervallo.

Infatti, se digitiamo:

```
PRINT 2.93873588E-39
```

il computer ripete $2.93873588E-39$, mentre con:

```
PRINT 2.93873587E-39
```

scrive 0.

Prima abbiamo visto che il computer ha eseguito l'operazione di raddoppio del più piccolo numero positivo di cui avverte l'esistenza. Mentre esso è ancora in memoria come valore attribuito ad A, digitiamo:

```
PRINT 7E - 39 + A - 7E - 39
```

il risultato è $2.93873588E-39$, come è ovvio attenderci;

```
PRINT 1+A-1
```

il risultato è 0, il che esce dalla logica aritmetica. Le risposte di cui sopra sono dovute a due motivi:

- quando un valore immesso nel computer viene incrementato, l'incremento deve superare un limite minimo affinché il computer si accorga che il valore considerato è stato aumentato;
- il limite minimo o incremento minimo deve essere proporzionale al valore immesso.

Lo conferma il fatto che, quando a $7E-39$ abbiamo aggiunto il numero rappresentato da A, il computer lo ha tenuto in considerazione: tale incremento infatti, benché piccolissimo, rispetto a $7E-39$ è proporzionalmente significativo ($7E - 9/2.93873588E - 39 = 2.38197657$); invece quando a 1 abbiamo aggiunto il medesimo numero il computer non lo ha recepito perché A è troppo piccolo rispetto a 1 ($1/2.93873588E - 39 = ?OVERFLOW ERROR$).

Poiché il computer lavora in aritmetica binaria, prendiamo in esame le potenze di due. Il C64 non può calcolare una potenza inferiore a 2^{-127} . Però se digitiamo PRINT LOG (2.93873588EE-39)/LOG(2) appare sul video -128.

Quindi $2.93873588E-39$ (primo e più piccolo numero positivo riconosciuto dal computer, nonché incremento minimo che occorre aggiungere a valori positivi di piccola entità affinché sia recepito il mutamento del loro ammontare) è una potenza di due, e precisamente 2^{-128} .

La struttura mnemonica del C64 e la numerazione binaria ordinata secondo le potenze di due danno origine alle seguenti caratteristiche:

- Ogni numero per diversificarsi da se stesso nella memoria del computer deve subire la miglioramento di almeno un incremento minimo equivalente a una potenza di due, proporzionale al numero in esame.
- L'incremento minimo è il limite iniziale di un arco (o sequenza o intorno) di numeri, il cui limite finale è dato dalla somma della potenza di due che rappresenta l'incremento minimo più la po-

Listato 1.

```

1 rem"incrim.minimo"
5 printchr$(144)
10 goto 1000
50 print"[clr]"
100 print"[down][right]ricerca dell' [rev on]incremento minimo[roff] per il"
110 input"[down][right]numero:";a
120 ifa<2.5243549e-29ora>7.92281624e+28then1510
150 print"[clr]"
160 forx=127to0step-1
170 q=-x:printq;b=2^q:printb;
180 c=a+b:printc;
190 ifa=cthenprint"u";c-a
200 ifc>athenprint"m";c-a
220 ifc>athen440
230 next
260 forx=0to126
270 q=x:printq;b=2^q:printb;
280 c=a+b:printc;
290 ifa=cthenprint"u";c-a
300 ifc>athenprint"m";c-a:goto440
310 next
440 print"[2down][5right](osserva gli ultimi risultati)"
450 print"[2down][6right]e poi premi [rev on]fi[roff]* * * * *)"
460 geta$:ifa$<>"[f1]"then460
480 print"[clr]":print"[down][right][rev on](a) valore scelto:";a
490 print"[down][right]incremento minimo valido per [rev on]a[roff] : "
492 k$=str$(b):k1$=left$(k$,11):k=val(k1$):k1=k+1e-08:k2$=str$(k1)
494 k3$=right$(k$,4):k4$=k2$+k3$:k2=val(k4$)
496 ifc>aanda+val(str$(b))=athenb=k2
500 print"[down][right]";b
510 printtab(10)"[down]* * * * *)"
520 print"[2down][right][rev on]a[roff] +";b;"- [rev on]a[roff] =";c-a
530 print
540 printa+b-a;"= n.ro rappresentativo"
550 print"[down][right]dell'arco (o intervallo) con inizio da:"
560 printb
570 printtab(12)"[down][right](premi [rev on]fi[roff])"
580 geta$:ifa$<>"[f1]"then580
710 q1=q+1
720 ls=2^q+2^q1
750 print"[clr]":print"valore scelto      ";a
760 print"[down][rev on]incrim.minimo[roff] rispetto al valore scelto"
770 print"e [rev on]inizio[roff] dell'arco      ";b
830 print"[down][rev on]n.ro rappresentativo[roff]"
840 print"dell'arco      ";a+b-a
850 d=val(str$(a+b-a))
859 ifls>=0.01andls<=1anda+ls-a>=dthenls=ls-1e-09:goto880
860 ifls>1andls<=10anda+ls-a>=dthenls=ls-1e-08:goto880
861 ifls>10 andls<=81anda+ls-a>=dthenls=ls-1e-07:goto880
862 ifls>81andls<=820anda+ls-a>=dthenls=ls-1e-06:goto880
863 ifls>=820andls<=10000anda+ls-a>=dthenls=ls-1e-05:goto880
864 ifls>10000andls<=99999anda+ls-a>=dthenls=ls-1e-04:goto880
865 ifls>=100000andls<=999999anda+ls-a>=dthenls=ls-1e-03:goto880
866 ifls>=1000000andls<=10000000anda+ls-a>=dthenls=ls-1e-02:goto880
867 ifls>=10000001andls<=100000000anda+ls-a>=dthenls=ls-1e-01:goto880
871 p$=str$(ls):p1$=left$(p$,11):p=val(p1$):p1=p-1e-08:p2$=str$(p1)
872 p3$=right$(p$,4)
873 ifls>=0.01andls<=1thenp3$="9"
874 ifls>=1000000000andls<=999999999thenp3$="9"
875 p4$=p2$+p3$
879 ifval(str$(a+val(str$(ls))-a))>dthen890
880 print"[down][rev on]fine[roff] dell'arco      ";ls
885 goto900
890 print"[down][rev on]fine[roff] dell'arco      ";p4$
900 print"[down]tutti i numeri compresi fra:"
901 ifval(str$(a+val(str$(ls))-a))>dthenprint:printb;" e ";p4$:goto905
902 print:printb;" e ";ls
905 print"[down]se sommati a";a"assumono"
907 print"[down]per il computer un [rev on]unico valore[roff]:"
908 print:printd
910 print"[down](premi [rev on]f3[roff] per continuare, [rev on]f5[roff] per finire)"
930 geta$:ifa$<>"[f3]"anda$<>"[f5]"then930
940 ifa$="[f3]"then50
950 ifa$="[f5]"then960
960 print"[clr]":end
1000 print"[clr]"
1010 printtab(9)"[5down]*****"
1020 printtab(9)"* * * * *"
1030 printtab(9)"* * incremento minimo * * * * *"
1040 printtab(9)"* * * * *"
1050 printtab(9)"* * e * * * * *"
1060 printtab(9)"* * * * *"
1070 printtab(9)"* * archi di numeri * * * * *"
1080 printtab(9)"* * * * *"
```

Continua a pag. 52

tenza immediatamente successiva. Occorre ricordare, come esempio relativo a potenze negative, che la potenza successiva a 2^{-31} è 2^{-30} .

- Tutti i numeri compresi entro il suddetto arco quando hanno funzione di incremento di uno specifico valore, non sono riconosciuti secondo il loro ammon-tare intrinseco, ma secondo quello di un numero che li rappresenta tutti.
- Il numero rappresentativo corrisponde al numero medio dell'arco.

Tenendo presente quanto sopra, possiamo stendere un programma che permetta di conoscere l'incremento minimo relativo a ogni valore. Volendo limitarci a un programma esemplificativo,

prenderemo in esame solo valori positivi. Diremo pertanto al computer di confrontare i valori positivi che gli verranno indicati con le potenze da 2^{-127} a 2^{-126} (rispettivamente la più piccola e la più grande potenza di due che il C64 possa calcolare), cioè inizialmente con un'entità minima di poco superiore allo zero, salendo a poco a poco fino a un numero composto per la parte intera di ben 38 cifre.

Il raffronto consisterà nel far aggiungere al valore indicato, che chiameremo A, una potenza di due di volta in volta crescente, e quindi nel far togliere dal risultato il medesimo A. Sin tanto che A più la potenza di due diminuito di A sarà ritenuto pari a zero dal computer, vorrà dire che nella sua memoria la potenza di due utiliz-

zata è una entità insignificante rispetto ad A. Non appena A più una potenza di due meno A produrrà un risultato diverso da zero, ciò significherà che il computer rispetto ad A considera tale potenza l'incremento minimo necessario a mutare nella propria memoria il valore di A. Tuttavia A maggiorato di tale incremento minimo potrebbe rimanere inalterato sul video, sul quale non possono essere riprodotti numeri superiori a nove cifre. Però nella memoria del computer invece dell'A visualizzato vi sarà un numero corrispondente ad A, maggiorato del numero rappresentativo dell'arco di cui l'incremento minimo è l'inizio.

Facciamo ricorso, nel tentativo di spiegarci meglio, proprio ai 600 milioni dell'esempio iniziale.

Dato l'avvio al programma, risulterà che l'incremento minimo che occorre aggiungere ai 600 milioni, affinché il computer si accorga del loro mutamento di valore, è 0.125 (2^{-3}). Infatti se digitiamo:

```
PRINT 600000000 + 0.125 - 600000000
```

il risultato non è 0,125 bensì 0,25 che è il numero rappresentativo dell'arco che parte da 0.125 (2^{-3}) e termina con 0.37499999; questo numero tende a 0.375, ossia alla somma delle due potenze consecutive 2^{-3} e 2^{-2} .

Il numero rappresentativo dell'arco è dato da $(0.125 + 0.375)/2 = 0.25$, ed è la risposta che il computer dà per tutti i numeri compresi nell'arco sopra precisato se sommati a 600 milioni, quindi anche per lo 0.2 e lo 0.37 che abbiamo utilizzato all'inizio delle presenti note. Infatti se digitiamo:

```
IF 600000000 + 0.2 = 600000000 + 0.37 THEN PRINT "SI"
```

il computer risponde SI, poiché nel presente caso ritiene 0.2 e 0.37 uguali a 0.25.

Una volta appresi i principi che governano l'incremento minimo, l'arco e il numero rappresentativo, proviamo a digitare:

Segue da pag. 51

```
1090 printtab(9) "*****"
1100 printtab(15) "[6down] (premi [rev on]fl[roff])"
1110 geta$:ifa$<>"[fl]"then1110
1120 print "[clr]"
1130 print "[down][right]l' [rev on]incremento minimo[roff] e' il numero piu'"
1140 print "[down][right]piccolo che, aggiunto ad un determinato"
1150 print "[right]valore, fa si' che il sistema operativo"
1160 print "[right]del computer recepisca che l'ammontare"
1170 print "[down][right]di quel valore e' mutato."
1180 print "[down][right]l' [rev on]incremento minimo[roff] e' proporzionale"
1190 print "[down][right]ai valori cui si riferisce ed e' anche"
1200 print "[down][right]l'inizio di un [rev on]arco[roff] (o sequenza o in="
1210 print "[down][right]torno) di numeri con caratteristiche"
1220 print "[down][right]particolari."
1230 printtab(15) "[2down] (premi [rev on]fl[roff])"
1240 geta$:ifa$<>"[fl]"then1240
1250 print "[clr]"
1260 print "[right]l' [rev on]arco[roff] ha inizio con l' [rev on]incremento mi="
1270 print "[down][right][rev on]nimo[roff], che e' una potenza di 2. la fine"
1280 print "[down][right]dell' [rev on]arco[roff] coincide con la somma della"
1290 print "[down][right]potenza corrispondente all'incremento"
1300 print "[down][right]minimo piu' la potenza immediatamente"
1310 print "[down][right]successiva."
1320 print "[down][right]tutti i numeri compresi nell' [rev on]arco[roff] assu="
1330 print "[right]mono per il computer un identico valo="
1340 print "[down][right]re, ossia, vengono identificati median="
1350 print "[right]te un [rev on]numero rappresentativo[roff]."
1360 printtab(15) "[2down] (premi [rev on]fl[roff])"
1370 geta$:ifa$<>"[fl]"then1370
1380 print "[clr]"
1390 print "[down][right]il [rev on]numero rappresentativo[roff] e' uguale al="
1400 print "[right]la somma dei limiti dell' arco divisa"
1410 print "[down][right]per 2."
1420 print "[down][right]- - - - -"
1430 print "[down][right]il programma esegue la ricerca dell' [rev on]in="
1440 print "[right][rev on]cremento[roff], dell' [rev on]arco[roff] e del [rev on]numero rap="
1450 print "[down][right][rev on]presentativo[roff] solo per numeri positivi"
1460 print "[down][right]compresi fra:"
1470 print "[down][4right]2.5243549e-29 e 7.92281624e+28"
1480 print "[2down][right] (premi [rev on]fl[roff] per utilizzare il programma)"
1490 geta$:ifa$<>"[fl]"then1490
1500 goto50
1510 print "[2down][2down][right]il programma accetta solo numeri posi="
1520 print "[down][right]tivi compresi fra:"
1530 print "[down][4right]2.5243549e-29 e 7.92281624e+28"
1540 printtab(15) "[5down] (premi [rev on]fl[roff])"
1550 geta$:ifa$<>"[fl]"then1550
1560 goto50
```


PRINT 600000000 + 0.4 - 600000000

La risposta è 0.5. Ma ormai è facile capire il perché. Dopo il primo arco di numeri, ne seguono altri. Rintracciati i dati del primo arco mediante l'utilizzo del programma, possiamo conoscere i dati degli archi successivi osservando alcune semplici regole:

- l'inizio dell'arco successivo è pari al numero cui tendeva il limite finale dell'arco precedente;
- il limite finale dell'arco successivo si trova aggiungendo al suo limite iniziale la stessa potenza aggiunta al limite iniziale dell'arco precedente, per ottenere di questo il limite finale;
- il numero rappresentativo è sempre dato dalla somma dei due limiti divisa per due.

Vale a dire che se il primo arco relativo a 600000000 ha i seguenti dati:

- limite iniziale: $0.125 = 2^{-3}$
- limite finale : 0.37499999 , tendente a $0.375 (2^{-3} + 2^{-2})$
- numero rappresentativo: $(0.125 + 0.375)/2 = 0.25$;

i dati dell'arco successivo sono:

- limite iniziale: 0.375
- limite finale: 0.62499999 , tendente a $0.625 (0.375 + 2^{-2})$
- numero rappresentativo: $(0.375 + 0.625)/2 = 0.5$

e così di seguito per gli archi ulteriori.

Torniamo al programma, il cui **listato** è pubblicato a pagina 45. Esso visualizzerà:

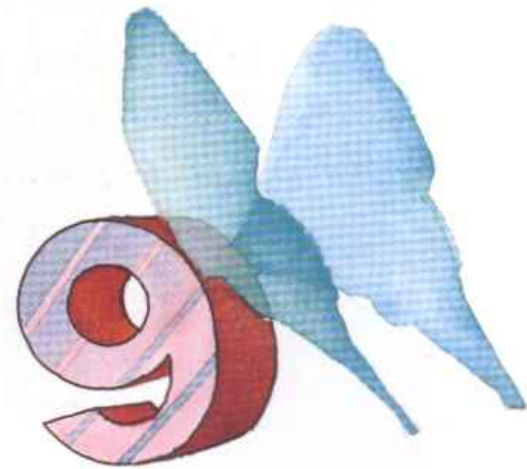
- l'esponente delle potenze di due utilizzate come incremento;
- le potenze di due;
- l'ammontare del valore scelto maggiorato delle potenze di due; tale ammontare rimarrà pari al valore scelto sino a quando la potenza di due utilizzata non sarà un incremento sufficiente ad alterarlo nella memo-

ria del computer, con l'avvertenza che, anche quando sarà sufficiente, l'ammontare visualizzato potrebbe restare inalterato se la potenza utilizzata inciderà solo sulle cifre esterne alla capienza del video;

- la lettera U, se il suddetto ammontare diminuito del valore scelto è per il computer uguale a zero; la lettera M, se è maggiore di zero;
- il risultato della seguente operazione: valore scelto, più la potenza di due, meno il valore scelto.

Non appena apparirà la lettera M il programma si fermerà. Premendo F1 esso indicherà, rispetto al valore introdotto, l'incremento minimo e il numero rappresentativo dell'arco. Premendo ancora F1 indicherà anche il limite finale dell'arco. Il listato è esteso, ma non complesso per chi abbia la paziente volontà di analizzarlo. Tuttavia occorre tener presente che:

- il frequente ricorso alle funzioni stringa VAL e STR\$ è dovuto alla necessità di far utilizzare al computer i numeri visualizzati depurati delle approssimazioni in memoria;
- il listato controlla l'esattezza dei limiti dell'arco risultanti dalle operazioni di confronto effettuate dal computer, e provvede alle opportune correzioni che possono rendersi necessarie sull'ultima cifra dei numeri che indicano i limiti; ciò avviene, con il ricorso ad alcuni artifici, alle linee da 492 a 496 e da 859 a 879;
- malgrado i controlli, le rettifiche e gli accorgimenti relativi, qualche cifra finale decimale potrebbe ancora risultare approssimata se proprio capitasse, nello svolgimento del programma, di aver a che fare con quei numeri che il computer non vuole assolutamente trascrivere esattamente. Per esempio, battete PRINT 6.98491944E-10 oppure PRINT 8.94069672E-08: il computer scriverà rispetti-



vamente 6.98491937 E-10 e 8.94069672E-08, ribellandosi a ogni tentativo di fargli ripetere il numero giusto;

- il programma che, come abbiamo già detto, effettua le ricerche unicamente per valori positivi, li accetta solo se compresi fra $2.5243549E-29$ e $7.92281624E+28$. Fuori di questo campo, peraltro vasto, il corso delle operazioni uscirebbe dai controlli consentiti dalla minima (2^{-127}) e massima (2^{126}) potenza di due, che il C64 è capace di utilizzare.

Ma giunti a questo punto che cosa ce ne facciamo di quanto il programma ci può dire? Niente, se la curiosità non ci sorregge. Altrimenti possiamo divertirci a immettere numeri a caso e scoprire che per il C64 (e con qualche differenza in altri computer) 100 milioni più 0.02 meno 100 milioni non fa 0.02, ma 0.03125; 10 miliardi più 5 meno 10 miliardi non fa 5, ma 4; 100 mila miliardi più 45000 meno 100 mila miliardi non fa 45000, ma 32768 e, per prendere un numero più piccolo, 1 più $3E-10$ meno 1 non fa $3E-10$, ma $4.65661287E-10$. Se poi oltre che dalla curiosità siamo animati dal desiderio di utilizzare il computer per calcoli matematici, il programma può servirci per tener conto del gioco delle approssimazioni.

Gustavo Fontanella

Sei concorrenti in gara per misurare la propria prontezza di riflessi, o magari per rispondere a un quiz...

Questo hardware, che potete realizzare con le vostre mani e azionare con il software contenuto nella cassetta, vi consentirà di intrattenere gli amici con giochi nuovissimi e divertenti.

E che vinca il più... velox!

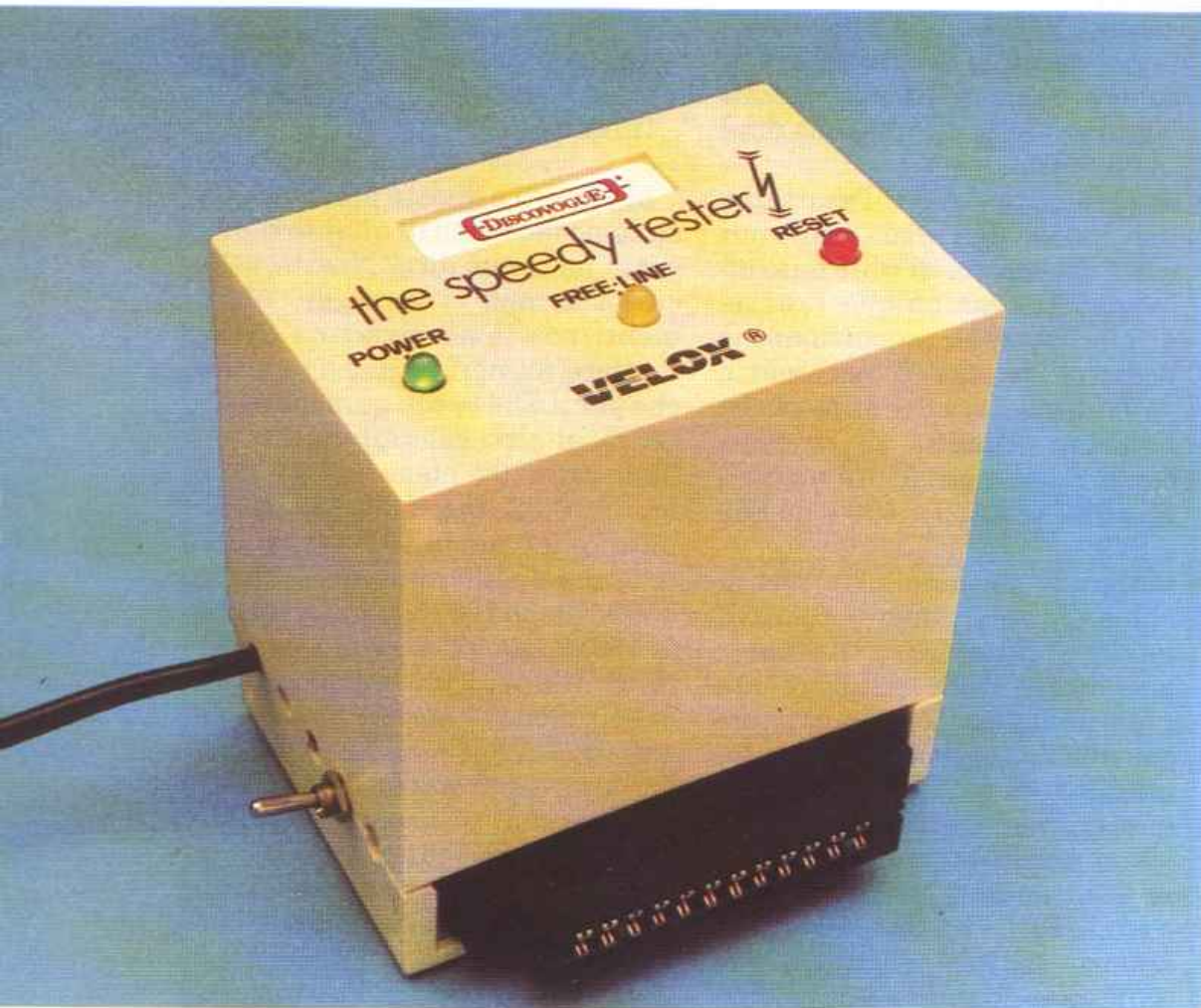
Velox è in grado di gestire in modo completo e professionale tutte le operazioni relative al controllo in contemporanea di più persone in dialogo diretto col computer tramite pulsanti periferici di azionamento, avendo a disposizione ben sette distinte linee indipendenti. In pratica è possibile ef-

fettuare, per esempio, un gioco a quiz con sei concorrenti che possono prenotarsi prima di rispondere, a loro volta coordinati da un supervisore arbitro del gioco con ruolo attivo sullo stesso circuito di comunicazione. Oppure si può organizzare una competizione tra più partecipanti per la verifica in

tempo reale della prontezza di riflessi. Il tutto sempre lasciando la tastiera del Commodore libera da manipolazioni e permettendo ai partecipanti interessati di non ostacolarsi a vicenda e di mantenersi a giusta distanza da computer e schermo.

Non esistono problemi di interfacciamento in quanto basta applicare Velox alla porta utente del computer stesso. Una sicura interfaccia ottica incorporata, realizzata con ben sette fotoaccoppiatori integrati, garantisce un totale isolamento tra apparecchio Velox e computer. Velox è dotato di una line-board duplicatrice che riporta sul retro del contenitore un'uscita uguale a quella già presente sul computer.

Il funzionamento avviene utilizzando programmi software più o meno complessi, in base alle prestazioni richieste e alle caratteristiche del supporto di memorizzazione (audio-cassetta o dischetto) da caricare sul computer utilizzato. Il monitor di segnalazione ottica è composto da un led verde lampeggiante (Power) che segnala con continuità all'utente la presenza della tensione di alimentazione e il corretto funzionamento di tutto il dispositivo, e da due led: uno giallo (Free line) e uno rosso (Reset) che, accendendosi, evidenziano all'istante rispettivamente la condizione di linea libe-



ra (senza prenotazioni né Reset) e l'azionamento, anche temporaneo, del sistema di Reset.

Analisi di funzionamento

Il circuito elettronico di Velox può essere innanzitutto diviso in due settori: uno che comprende lo stadio alimentatore e il raccordo di connessione alla porta utente del computer, relativo al circuito stampato carrier (cod. 100.57), e un altro esecutivo e di controllo relativo invece al circuito stampato master (cod. 149.66).

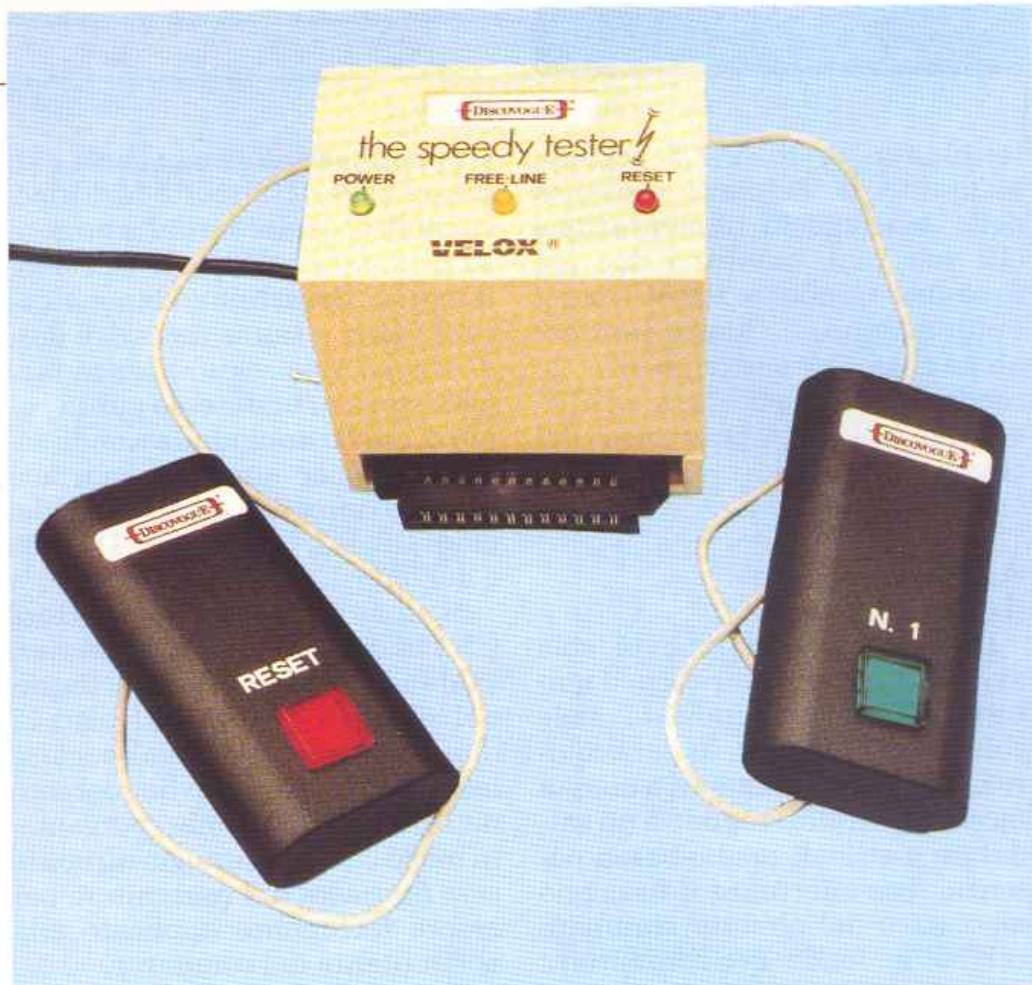
Il primo settore fornisce al secondo l'alimentazione per il funzionamento e provvede a trasferire i segnali da elaborare sul databus della porta utente (in input).

Un potente alimentatore in c.c. abbassa la tensione alternata di 220 volt presente sul primario del trasformatore TF1 (pin 1 e 4) a circa 15 volt (pin 9 e 12 del secondario) per poi raddrizzarla a circa 22 volt c.c. (ponte diodi D1) e filtrarla con C1, IC1 e C2, stabilizzandola a 12 volt costanti in corrente continua, disponibili su due pin della linea di connessione all'altro circuito stampato (simboli grafici positivo e massa).

Anche la tensione di 220 volt viene riportata su tre pin della linea di connessione (simboli grafici: sinusoidale, terra e sinusoidale) con tanto di linea di terra per eventuali collegamenti.

L'interruttore SW1, collegato ai punti D ed E del circuito, consente di accendere e spegnere lo stadio alimentatore, permettendo o impedendo il passaggio della 220 volt in arrivo dal cavetto di alimentazione SP1 collegato alla rete. Alla linea di connessione sono riportati anche i 24 pin che consentono al computer collegato di comunicare in input con il circuito master di Velox, e in input e/o in output con altre eventuali periferiche.

Il secondo settore è relativo al circuito stampato master ed è il più importante: lavora infatti in diretto abbinamento al computer,



elaborando e trasformando i segnali di attivazione provenienti dai pulsanti di controllo periferici. Il collegamento con la linea dati della porta utente è realizzato attraverso i nove terminali 1, 2, C, D, E, F, H, J, e K, ovvero 0 volt (1), 5 volt positivi (2), PB0 (C), PB1 (D), PB2 (E), PB3 (F), PB4 (H), PB5 (J) e PB6 (K): si tratta di una connessione soltanto ottica, in quanto è presente un'interfaccia realizzata con i sette fotoaccoppiatori IC3, IC4, IC5, IC6, IC7, IC8 e IC9.

Durante il funzionamento attivo del software di gestione, nelle condizioni iniziali del sistema, cioè quando è stato inviato un Reset, se nessuno dei pulsanti S1, S2, S3, S4, S5, S6 ed S7 viene premuto, tutti gli ingressi dei fotoaccoppiatori di interfacciamento (pin 1) sono disattivati, presentandosi all'ingresso uno stato logico zero (nullo per l'azione delle resistenze R11, R13, R15, R17, R19, R21 ed R23 collegate a massa). Di conseguenza sulle linee di input a valle degli integrati stessi (pin 5) è mantenuto un segnale logico di Reset (positivo logico corrispondente alla configurazione binaria 0111/1111, ovvero decimale 127).

Nessun segnale giunge alla porta invertente NAND IC2b, che mantiene acceso, tramite la resistenza R10, il led giallo L2 free-line segnalatore di linea di comunicazione completamente libera.

Se viene premuto il pulsante di Reset S1 (Reset), allora un segnale positivo viene inviato al pin 1 di IC3 e si causa di conseguenza una commutazione di stato logico su PB6; si accende, tramite R12, il led rosso L3 Reset che evidenzia l'attivazione e si spegne, per l'azione di D2 e della porta invertente IC2b, il led L2 free-line quando spento, segnala che la linea non è più completamente libera, ovvero vi sono uno o più pulsanti sono premuti. Premendo invece i pulsanti numerati S2, S3, S4, S5, S6 ed S7 la commutazione logica si manifesta, per lo stesso principio di funzionamento, sugli altri corrispondenti fotoaccoppiatori collegati in configurazione modulare, con la differenza che il led giallo L2 si spegne per l'azione dei diodi, mentre anche L3 di colore rosso rimane spento (Reset non attivato).

Il led verde L1 (Power) pilotato dall'oscillatore IC2a tramite T1, lampeggiando alla frequenza de-

terminata da R2 e C3 segnala all'utente la regolare presenza della tensione di alimentazione (12 volt c.c.), mentre i due led Free-line (giallo) e Reset (rosso) evidenziano, accendendosi all'istante, rispettivamente le condizioni di linea libera e l'attivazione anche temporanea del circuito di Reset.

Assemblaggio del circuito

Il montaggio va iniziato solo avendo a disposizione tutto il materiale originale indicato nell'elenco componenti, in particolare i due circuiti stampati a doppia faccia. Oltre allo stagno e a un saldatore a stilo di medio-bassa potenza, per effettuare il fissaggio meccanico del circuito stampato carrier al fondo del contenitore tramite quattro viti serve solo un cacciavite a croce.

Occorre evitare di causare so-

vrapposizioni di componenti e di creare zone del circuito rese inaccessibili al saldatore dalla presenza di componenti come TF1 o gli integrati. Si raccomandano saldature veloci e senza eccessive dosi di stagno, da eseguire senza mai persistere su uno stesso punto per più di quattro o cinque secondi, soprattutto per quanto riguarda i pin degli integrati e i terminali dei diodi D2, D3, D4, D5, D6, D7 e D8 e dei led L1, L2 ed L3.

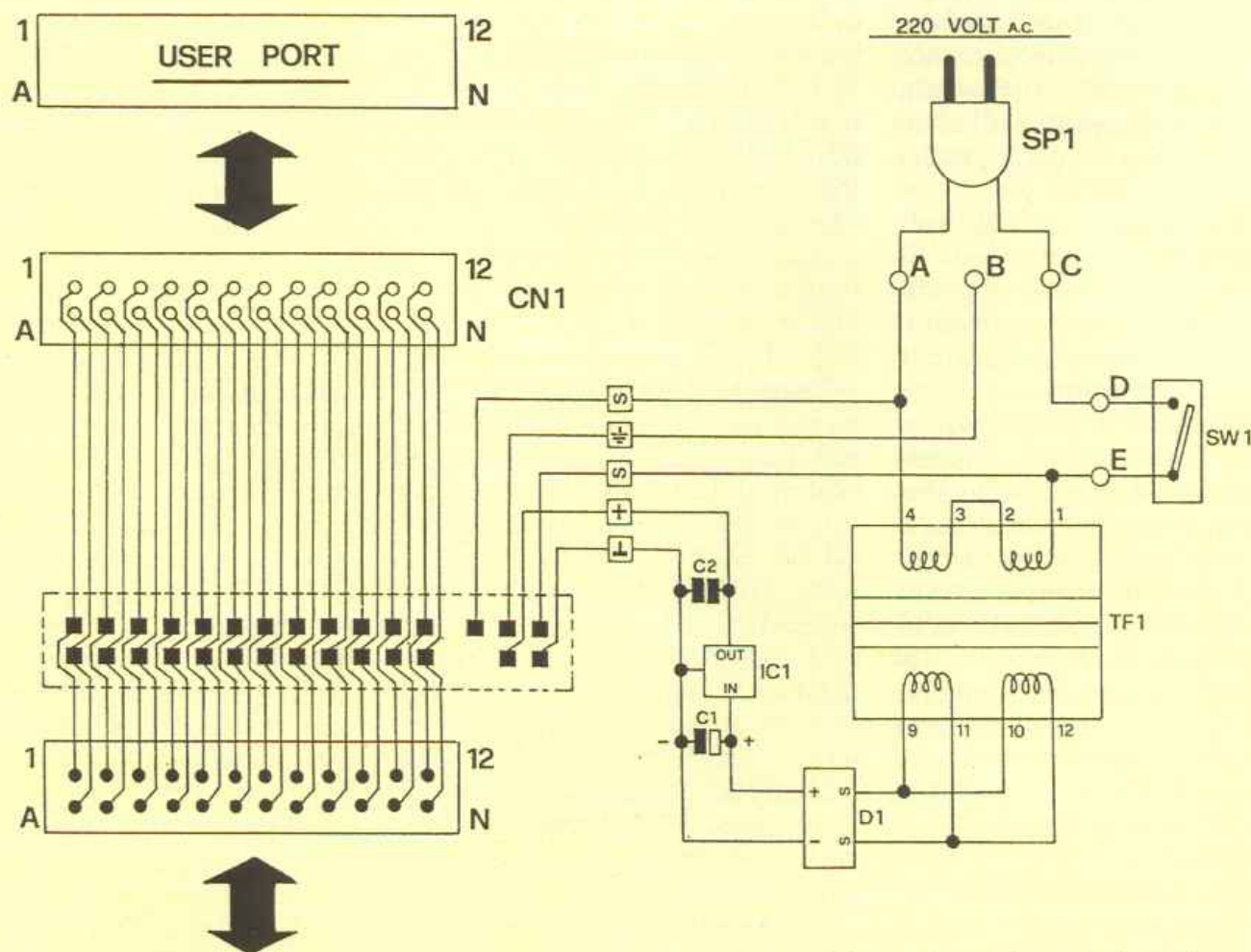
Si inizia montando il connettore CN1 al circuito stampato carrier, saldando la prima fila di 12 terminali al lato A e la fila degli altri 12 al lato B: l'operazione va eseguita lasciando il corpo plastico del connettore il più possibile sporgente, in modo che a montaggio ultimato esso fuoriesca dal contenitore per essere facilmente applicato alla porta utente del computer.

Si può quindi procedere alla stagnatura dei 29 terminali che formano la linea di connessione al

circuito stampato master, il quale sarà successivamente montato proprio lungo la linea indicata dalla serigrafia, e cioè tra le due file di terminali.

In seguito occorre montare sul lato A (quello superiore) i cinque chiodini capicorda ai punti contrassegnati con A, B, C, D ed E, e quindi i condensatori C2 e C1 (quest'ultimo con il terminale positivo verso il chiodino capicorda C): tutte le saldature vanno effettuate sul lato opposto B. A questo punto si può montare, sempre sul lato A e sempre saldando sul lato opposto B, il trasformatore TF1 (dotato di ben dieci terminali che da soli consentono un ottimo fissaggio anche meccanico) che, con il suo notevole volume, riempie di colpo quasi tutto lo spazio disponibile sul lato A.

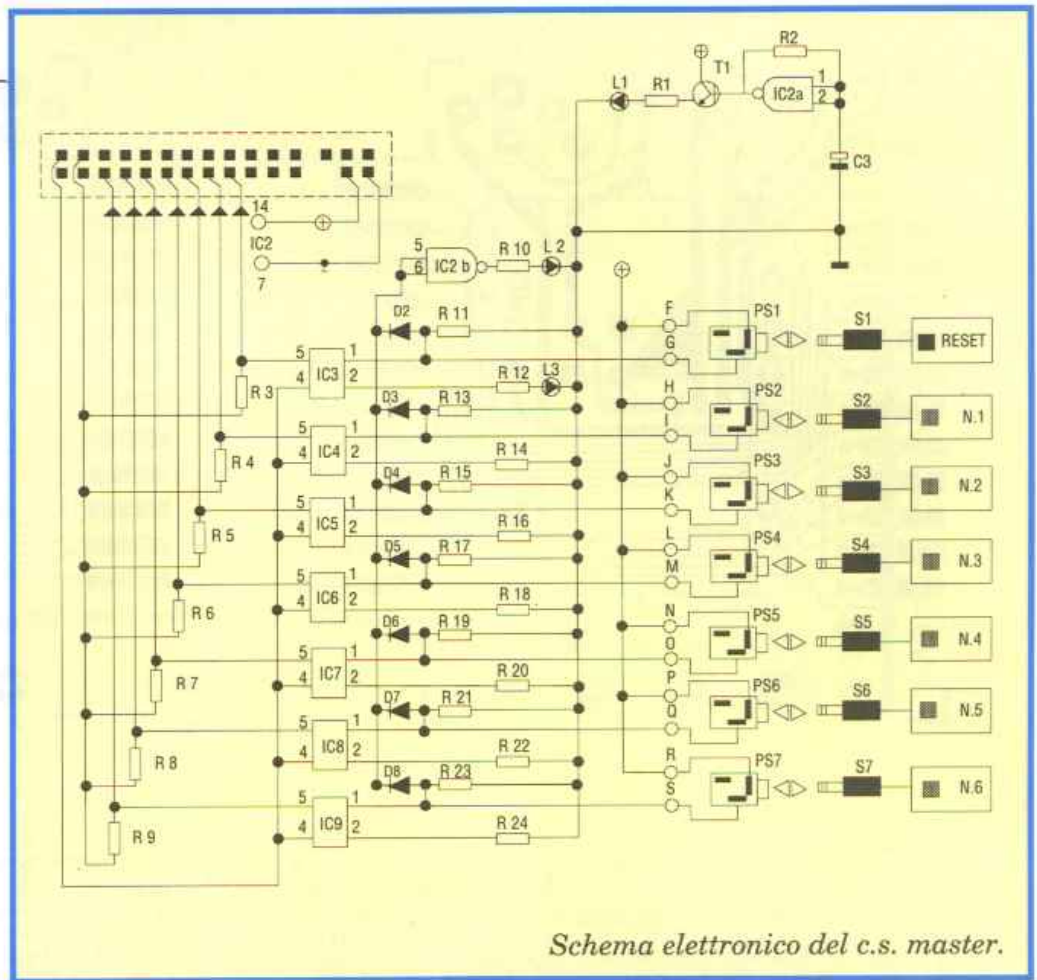
Sul lato B vanno montati il ponte diodi D1, in modo che il corpo nero rimanga sollevato il meno possibile, e il regolatore di tensio-



Schema elettronico relativo al c.s. carrier.

ne IC1 da piegare poi di 90° facendo perno sui tre terminali affinché non sia d'ingombro sporgendo troppo: è già previsto un apposito spazio riservato all'aletta metallica che in questo modo arriva a sfiorare la superficie del circuito stampato. Queste saldature sono da effettuarsi, per entrambi i componenti, sullo stesso lato B. Il circuito stampato carrier può a questo punto essere messo momentaneamente da parte: si prosegue con il master stagnando i 29 terminali della linea di connessione (14 sul lato A e 15 sul lato B) e montando, sul lato A, i 14 chiodini capicorda rimanenti (ai punti contrassegnati con F, G, H, I, J, K, L, M, N, O, P, Q, R ed S) e, nell'ordine, i seguenti componenti (con saldature da effettuarsi sul lato opposto B): circuito integrato IC2; transistor T1; resistenze R1 ed R2; condensatore C3 (terminale positivo verso IC2); resistenze R10 ed R12; circuiti integrati IC3, IC4, IC5 e IC6; diodi D2, D3, D4 e D5; resistenze R11, R13, R14, R15, R16, R17 ed R18; circuiti integrati IC7, IC8 e IC9; diodi D6, D7 e D8; resistenze R19, R20, R21, R22, R23 ed R24.

Si prosegue quindi con il montaggio sul lato B dei restanti componenti (saldature da effettuarsi



Schema elettronico del c.s. master.

sul lato opposto A oppure, dove ciò non sia possibile, sullo stesso lato B).

Nell'ordine: resistenze R3 ed R4; resistenze R5, R6 ed R7; resistenze R8 ed R9.

Il montaggio va ultimato con i diodi led L1, L2 ed L3 da fissare sul lato A (saldature sul lato opposto B), con i terminali piegati di 90° in modo che le incapsulature

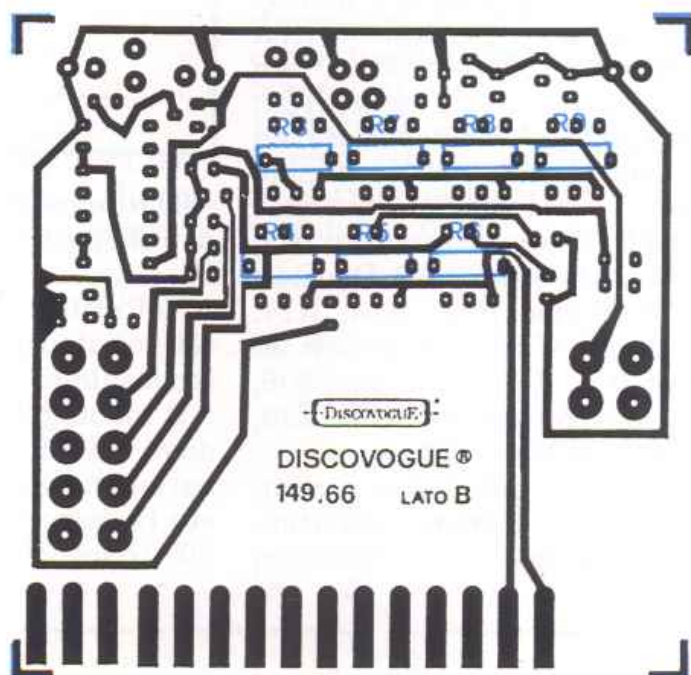
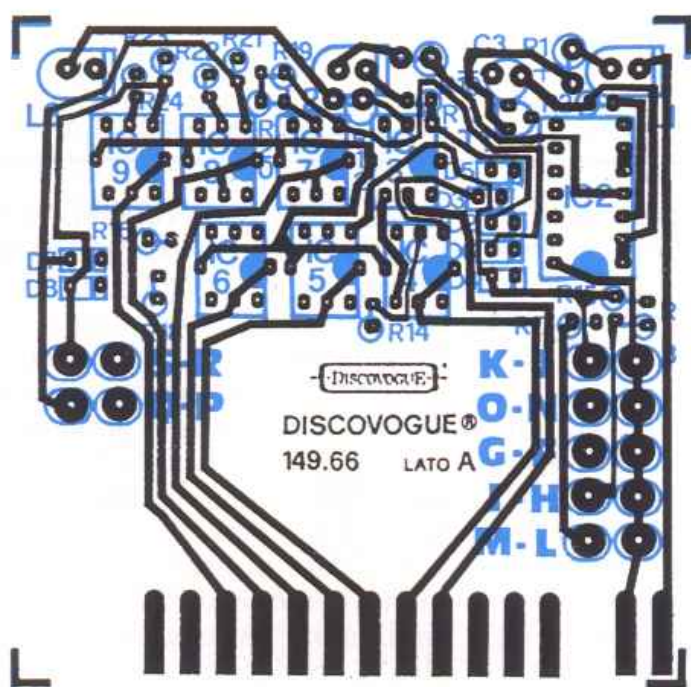
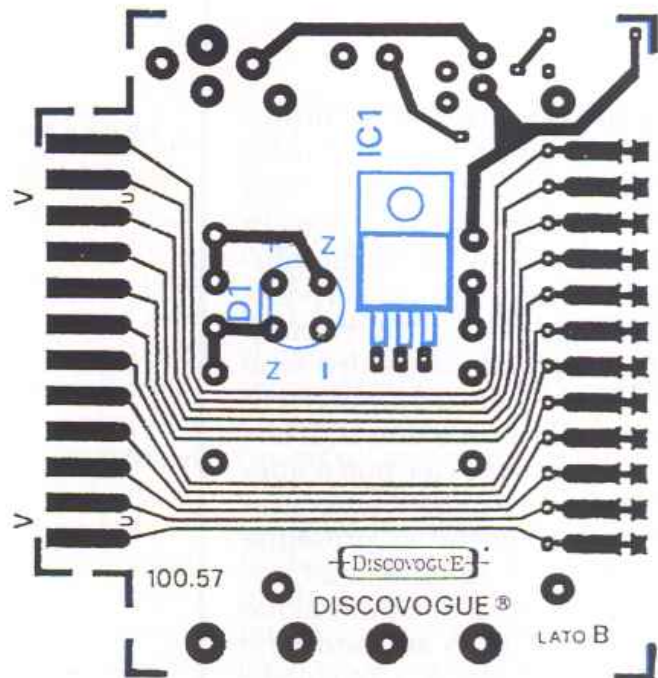
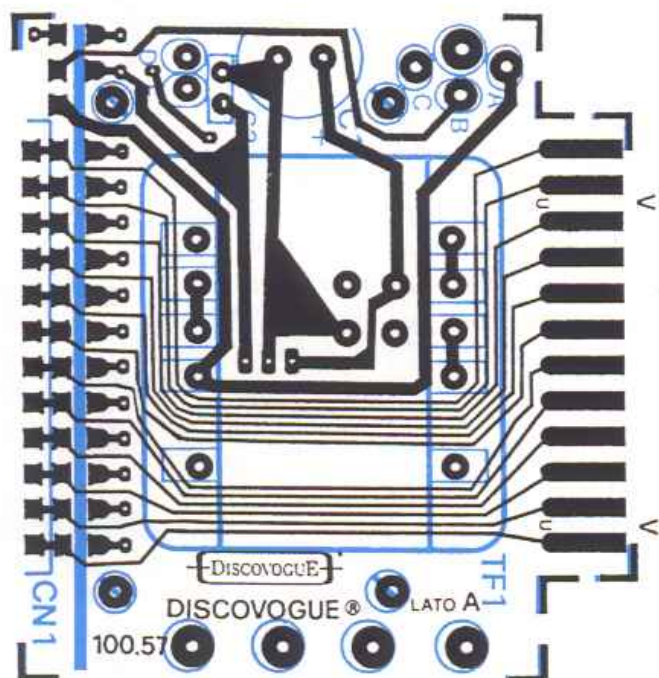
di colore verde (L1), giallo (L2) e rosso (L3) risultino parallele al piano del circuito stampato, distanziate di circa un centimetro e sporgenti dal relativo bordo esterno di due o tre millimetri, così da fuoriuscire poi dal pannello del coperchio del contenitore attraverso gli appositi fori. Come indicano le serigrafie del circuito stampato

master, per tutti i componenti da montare sul lato A (tranne che per i diodi D2, D3, D4, D5, D6, D7 e D8) è previsto il montaggio assiale perpendicolare al piano circuitale: questo sistema unito alla piena utilizzazione delle superfici disponibili (lato A e lato B) ha permesso di ridurre al minimo l'ingombro di Velox.

A questo punto si effettua l'unione dei due circuiti stampati saldando i 29 terminali del master ai corrispondenti 29 del carrier: si ottiene così un insieme molto compatto e resistente. Le saldature di connessione hanno una funzione meccanica, oltre che elettrica, perciò dovranno essere tecnicamente perfette, precise e realizzate abbondando con lo stagno. I due stampati risultano tra loro perpendicolari e saldamente uniti lungo la linea di connessione, con il lato A rivolto verso il trasformatore TF1 e il lato B rivolto verso il connettore CN1, cioè verso l'esterno dell'insieme.

PULSANTE PREMUTO	USER PORT	LED	
		FREE-LINE	RESET
NESSUNO	127	☀	●
N. 1	95	●	●
N. 2	111	●	●
N. 3	119	●	●
N. 4	123	●	●
N. 5	125	●	●
N. 6	126	●	●
RESET	63	●	☀

Ultimato il montaggio preliminare l'insieme può essere inserito nell'apposito contenitore, fissando il circuito stampato carrier al fondo tramite le quattro piccole viti da inserire nei relativi pila-



Piste conduttrici e montaggio componenti del lato A e B del c.s. carrier in alto e del c.s. master in basso.

strini di sostegno: il connettore CN1 spoggerà dall'apertura anteriore, mentre D1 e IC1 rimarranno sollevati di qualche millimetro dal fondo.

Prima di chiudere il contenitore occorre ovviamente effettuare tutti i collegamenti. L'interruttore SW1 va saldato ai due chiodini dei punti circuitali D ed E del circuito stampato carrier, in modo che possa fuoriuscire con la sua parte anteriore (levetta di azionamento e filettatura metallica) dal secondo foro anteriore presente sul lato sinistro del contenitore.

Analogamente va saldato il ca-

vetto di alimentazione SP1 che dovrà entrare dall'ultimo foro posteriore sinistro del contenitore: va infilato (dal sotto e verso l'alto) nell'apposito punto di passaggio previsto sullo stampato tra i tre punti circuitali A, B e C in modo che fuoriesca sul lato A e possa così essere stabilmente convogliato e saldato ai due chiodini dei punti A e C.

Il chiodino centrale (punto B) predisposto per la linea di terra rimane libero in quanto non serve al funzionamento di Velox.

I collegamenti del circuito master prevedono invece l'allaccia-

mento delle sette prese jack alle rispettive coppie di punti circuitali F e G, H e I, J e K, L ed M, N e O, P e Q e infine R ed S: ogni presa ha tre piccoli terminali, che sono nell'ordine massa (quello centrale vicino alla filettatura da collegare ai punti F, H, J, L, N, P ed R), segnale supplementare (terminale distanziato dagli altri due, da non collegare) e segnale dei pulsanti periferici (rimanente terminale, da collegare ai punti G, I, K, M, O, Q ed S).

PS1, PS2, PS3, PS4, PS5, PS6 e PS7 sono compatibili con gli spinotti jack dei sette pulsanti S1,

S2, S3, S4, S5, S6 ed S7 da collegare a Velox.

Collaudo e installazione

Terminate le operazioni di collegamento è consigliabile eseguire con un tester un semplice collaudo, verificando la presenza della 220 volt a.c. sui punti A e C del circuito carrier e sull'avvolgimento primario di TF1 (pin 1 e 4): ovviamente questo andrà fatto con il tester predisposto sulla corrente alternata (1.000 volt f.s. a.c.) e collegando il circuito alla rete tramite SP1 con l'accensione dell'interruttore SW1.

Analogamente sul secondario di TF1 dovrà essere presente una tensione di circa 15 volt (tester a 50 volt f.s. a.c.) mentre, con lo strumento predisposto sulla corrente continua (50 volt f.s. c.c.), sui terminali di C1 si dovrà riscontrare un potenziale di circa 21 volt che scende a 12 volt a valle di IC1 e sui terminali di C2. Lo stesso valore di 12 volt c.c. deve essere presente sul circuito master, più precisamente sul pin 14 di IC2, sul collettore del transistor T1 e sui punti circuitali F, H, J, L, N, P ed R. Occorre poi rilevare il

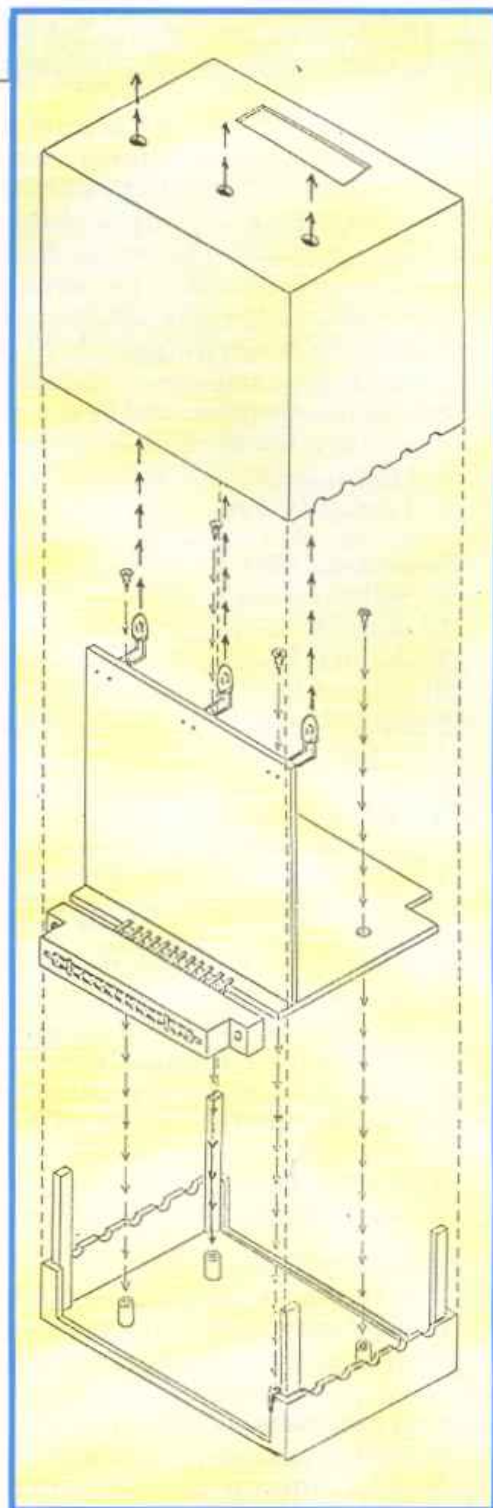
corretto funzionamento dell'oscillatore IC2a rilevando sul pin 3 un segnale di circa 3 Hz., valore determinato da R2 e C3. Se le misure danno esito positivo si può passare alla verifica finale pratica collegando Velox alla porta utente del computer tramite il connettore CN1, che in parte fuoriesce dalla finestra anteriore del contenitore.

Quindi si inseriscono gli spinotti dei cavetti dei sette pulsanti periferici S1, S2, S3, S4, S5, S6 ed S7 nelle relative prese jack PS1, PS2, PS3, PS4, PS5, PS6 e PS7 appositamente predisposte, avendo cura di non creare falsi contatti premendo inavvertitamente i pulsanti stessi (di colore rosso S1, verde gli altri).

Le suddette operazioni vanno ovviamente eseguite con gli apparecchi spenti, da accendersi solo successivamente.

Se la verifica finale fornisce esito positivo il circuito, già inserito nel fondo del contenitore, può essere definitivamente chiuso con il coperchio, da far scorrere sulle quattro sporgenze-guida angolari che si ergono dal fondo, in modo che alla fine i tre led sporgano correttamente dai corrispondenti fori presenti sul pannello frontale

superiore serigrafato, e che l'interruttore SW1 e il cavetto di alimentazione SP1 fuoriescano dai fori laterali di sinistra (SP1 dall'ultimo in fondo, SW1 dal secondo anteriore). Le prese jack PS1, PS2, PS3, PS4, PS5, PS6 e PS7 vanno invece sistemate sul retro del coperchio, negli appositi sette fori previsti. Alle sporgenze filettate vanno applica-

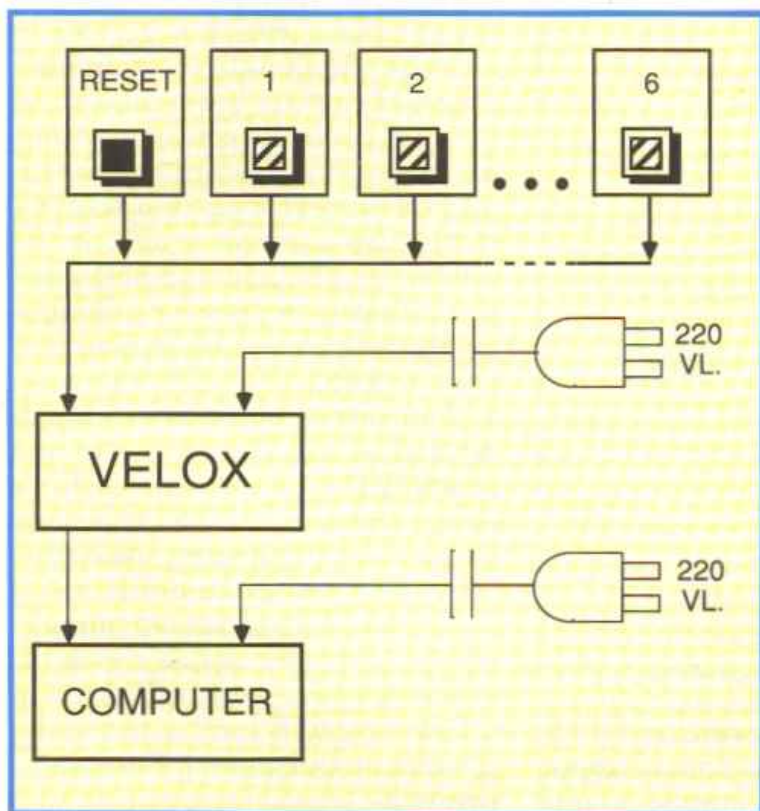


ti i piccoli dadi metallici di fissaggio. Anche alla sporgenza filettata dell'interruttore SW1 va applicato il piccolo dado metallico di fissaggio.

Uso ottimale del dispositivo

Velox consente un dialogo multilinea tra più utenti e computer senza che sia necessario usare la tastiera: questo è possibile utilizzando speciali pulsanti collegati via filo al circuito elettronico dell'apparecchio stesso.

La capacità massima è relativa alla configurazione di sei pulsanti di azionamento e di un pulsant-



Elenco componenti

L'hardware Velox è composto in prevalenza da circuiti integrati, perciò componenti elettronici specifici come resistenze e condensatori sono presenti in limitata quantità, relativamente alle caratteristiche dell'apparecchio. Questo è a vantaggio della sicurezza operativa e soprattutto è garanzia di affidabilità e perfetto funzionamento. La parte elettro-meccanica comprende tra l'altro due circuiti stampati, un connettore, sette prese jack, il contenitore plastico forato e serigrafato e i sette speciali pulsanti di controllo periferici completi di cavetto e spinotti jack.

Nell'elenco componenti di seguito riportato è possibile distinguere: quantità di ogni tipo e valore di articolo (numeri tra parentesi); eventuali codici di circuiti; eventuali valori espressi nell'unità standard di misura. I limiti massimi di tolleranza si intendono 5% per le resistenze e 10% per i condensatori.

Resistenze (24)

- (1) R1: 560 ohm 1/2 W
- (1) R2: 120 Kohm 1/4 W
- (7) R3, R4, R5, R6, R7, R8 ed R9: 10 Kohm 1/4 W
- (2) R10 ed R12: 120 ohm 1/2 W
- (13) R11, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23 ed R24: 1 Kohm 1/4 W

Condensatori (3)

- (1) C1: 470 microF 35 VL elettrol. vert.
- (1) C2: 100 nanoF 100 VL poliest.
- (1) C3: 4,7 microF 63 VL elettrol. vert.

Diodi (11)

- (1) D1: W02 ponte
- (1) L1: led rotondo 5 mm. colore verde
- (1) L2: led rotondo 5 mm. colore giallo
- (1) L3: led rotondo 5 mm. colore rosso
- (7) D2, D3, D4, D5, D6, D7 e D8: 1N4148

Transistor (1)

- (1) T1: BC546A

Circuiti integrati (9)L

- (1) IC1: 7812 regolatore di tensione
- (1) IC2: 4093 quad NAND Schmitt trigger
- (7) IC3, IC4, IC5, IC6, IC7, IC8 e IC9: MCT2E fotoaccoppiatore

Altri componenti (18)

- (1) TF1: trasformatore 220/15 VL 400 mA
- (1) SW1: interruttore unipolare miniatura
- (1) SP1: cavetto di alimentazione 220 VL con spina
- (7) PS1, PS2, PS3, PS4, PS5, PS6 e PS7: presa jack stereo da pannello 3,5 mm.
- (1) S1: telecomando con pulsante di controllo colore rosso, completo di cavetto bipolare di collegamento con spinotto jack stereo 3,5 mm.
- (6) S2, S3, S4, S5, S6 ed S7: telecomando con pulsante di controllo colore verde, completo di cavetto bipolare di collegamento con spinotto jack stereo 3,5 mm.
- (1) CN1: connettore 12 + 12 poli passo 3,96 mm.

Vari (33)

- (1) circuito stampato cod. 100.57
- (1) circuito stampato cod. 149.66
- (4) viti di fissaggio per c.s.
- (19) chiodini terminali capicorda per c.s.
- (7) tranci piattina bipolare lung. cm. 8
- (1) contenitore plastico forato e serigrafato cod. 149.21

corretta distanza dal computer e dal video.

I pulsanti di controllo sono collegabili, tramite normali spinotti jack, direttamente, e nella quantità desiderata, sul retro all'apparecchio Velox: quelli di colore verde sono numerati da uno a sei e sono per i singoli giocatori, quello di colore rosso è contrassegnato con Reset e genera l'impulso di comando per il ritorno alle condizioni iniziali (resettaggio, azzeramento). L'azionamento è semplice perché basta premere il pulsante tenendo in mano il piccolo involucro plastico nero che lo contiene e protegge: il filo di collegamento è sufficientemente lungo (80 centimetri) per permettere un posizionamento ideale a ogni persona.

Nel caso non sia disponibile un apposito supervisore (arbitro di gioco, controllore) per il pulsante di Reset, questo sarà usato da chi, tra i presenti, viene abilitato a farlo (e avrà dunque due pulsanti), oppure sarà tenuto a disposizione di tutti in posizione facilmente accessibile (di fianco alla tastiera del Commodore).

Software dimostrativo

Questo programma, pur non contenendo complesse routine di elaborazione, riesce a gestire in modo professionale e totalmente automatico tutte le operazioni relative al controllo contemporaneo di più giocatori in gara tra loro (fino a un numero massimo di sei distinti); ciò avviene selezionando quello che per primo rispetto agli altri effettua la pressione del proprio pulsante elettronico di prenotazione (con esclusione immediata degli altri e senza possibilità di errore) e consentendo dunque competizioni di prontezza di riflessi, giochi a quiz con selezione automatizzata oppure qualsiasi altro sistema di comunicazione multilinea a distanza tra più persone e computer, senza che la tastiera debba essere utilizzata a questo scopo. È inoltre possibile, tramite un pulsante di Reset indipendente (gestibile anche da una

te di sovracontrollo, per un totale di sette linee indipendenti: in pratica sette persone possono parte-

cipare attivamente senza usare la tastiera, senza avvicinarsi troppo reciprocamente e stando a una

Come si acquista il materiale già pronto

È disponibile la versione hardware, ovvero l'apparecchio già montato, collaudato e funzionante, completo di istruzioni di installazione e uso. Codice 149.00, lire 121.000. Chi ha un minimo di esperienza con l'elettronica e col saldatore può acquistare la versione hardware kit, una scatola di montaggio completa comprendente, oltre a tutto il materiale indicato nell'elenco componenti, anche le istruzioni di assemblaggio, collaudo, installazione e uso. Codice 149.10, lire 97.500.

È inoltre possibile richiedere il Personal set (una confezione comprendente solo i due circuiti stampati, il connettore, il contenitore, i relativi accessori di fissaggio e le istruzioni di assemblaggio, collaudo, installazione e uso) per costruire l'apparecchio Velox avendo già a disposizione il rimanente materiale necessario, oppure per realizzare periferiche personali diverse tutte dotate di duplicazione della porta utente sul retro che consentono il collegamento in cascata di altre periferiche addizionali. Codice 149.20, lire 45.500.

Per quanto riguarda il software esistono diverse versioni di programmi per computer Commodore 64 e Commodore 128, da utilizzare in base alle prestazioni desiderate e alle caratteristiche del supporto di memorizzazione (cassetta o dischetto):

- Software dimostrativo su cassetta (**allegato a questo numero di RE&C**), semplice, ma versatile e potente, che permette di gestire in modo professionale e totalmente automatico tutte le operazioni relative al controllo contemporaneo di più giocatori (massimo sei) con pulsanti per prenotazioni a risposte su quiz o per verifiche di prontezza di riflessi. Possibilità di usare il programma senza l'uso della tastiera del computer. Sistema di Reset automatico tramite apposita linea di azionamento. Monitoraggio completo degli eventi, de-

gli errori comportamentali e dei responsi forniti in tempo reale. Generazione di un tempo casuale di attesa variabile. Codice 149.61, lire 8.000.

- Software di funzionamento su cassetta (**che sarà allegato al prossimo numero di RE&C**), che presenta le stesse caratteristiche del programma precedente, ma con videata grafica multicolor di supporto e routine supplementare per la gestione specifica di una gara di risposte a quiz forniti automaticamente dal computer. Gestione del punteggio di ognuno dei sei (numero massimo) concorrenti. Sistema automatico di selezione della risposta ritenuta corretta tramite lo stesso pulsante di prenotazione. Codice 149.62, lire 26.000.

- Software di funzionamento su dischetto che presenta le stesse caratteristiche del programma cod. 149.62, ma con possibilità di gestione specifica, oltre che del gioco a quiz, anche di una competizione per la verifica della prontezza di riflessi (massimo 6 partecipanti) e di un divertente gioco per la creazione di coppie ideali (massimo tre coppie di partecipanti). Codice 149.63, lire 42.500.

Tutti gli ordini d'acquisto possono essere effettuati tramite lettera da indirizzare in busta chiusa a: Discovogue - P.O. BOX 495 - 41100 Modena.

I prezzi comprendono l'Iva, con pagamento contrassegno e spese di spedizione a carico del destinatario. Gli invii si effettuano ovunque, entro 24 ore dall'arrivo dell'ordine, tramite pacco postale che a richiesta può essere anche urgente (con maggiorazione delle spese aggiuntive).

Ogni ordine dà diritto a ricevere in omaggio, oltre a una piacevole sorpresa, anche la mailing card personalizzata e codificata che consente di ottenere sconti e agevolazioni in eventuali ordini successivi.

persona-arbitro), annullare prenotazioni irregolari o comandarne altre, sempre senza usare la tastiera del computer.

In pratica il software è in grado di elaborare tutti i segnali impulsivi che, attraverso i sette input PB0, PB1, PB2, PB3, PB4, PB5, PB6 e PB7 della porta utente del computer, determinano la configurazione elettronica di riferimento che si determina di volta in volta (prenotazioni singole, multiple, con o senza Reset). Non appena il programma ha inizio appare sullo schermo la videata-monitor a fondo verde, bordo verde e scritte nere, con il nome Velox 1 e il data-set di copyright nella parte alta dello schermo: più sotto viene proiettato il display (delimitato da due linee) contenente il messaggio di responso per l'utente che inizialmente è "Per iniziare attivare il Reset". Occorre infatti premere il pulsante Reset (quello rosso) per iniziare la fase esecutiva del programma. A fondo schermo il messaggio permanente "Attenzione attivare sempre il Reset prima di effettuare ogni gara" ricorda invece che lo stesso pulsante

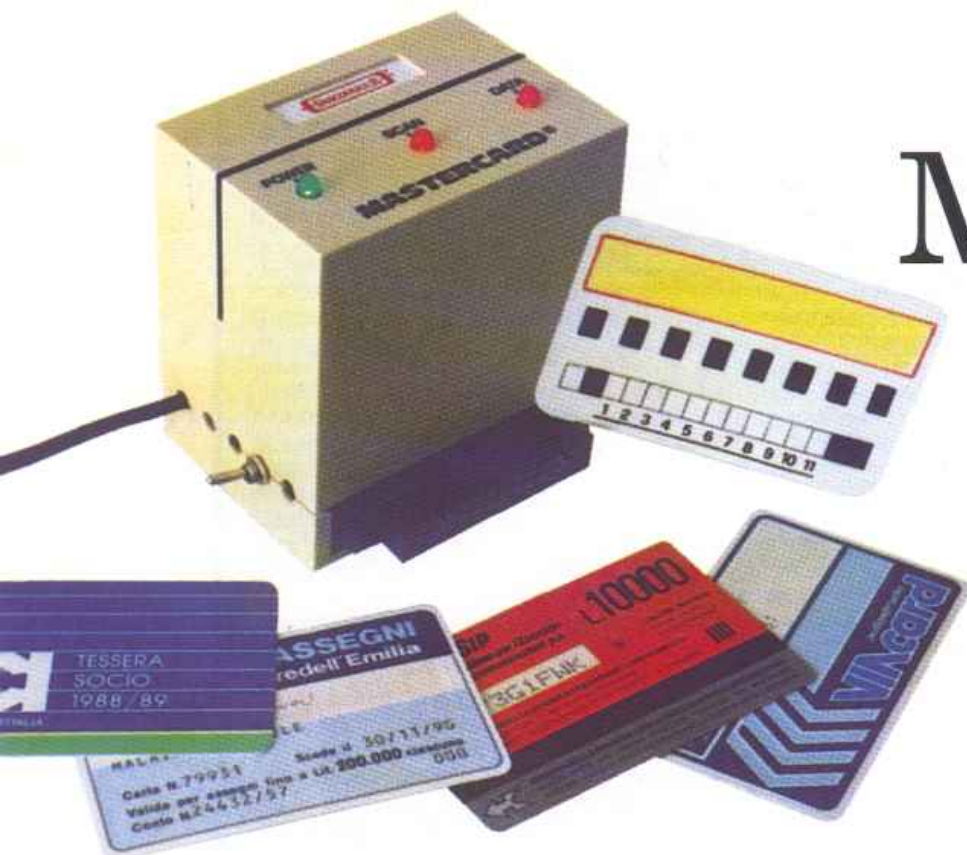
Reset va premuto ogni volta che si desidera far ritorno al punto di inizio del programma stesso. Dopo aver premuto il pulsante Reset il display visualizza il messaggio "Pronti per la risposta..." per un tempo di attesa casuale compreso tra un minimo di un secondo e un massimo di quattro secondi: durante questo tempo nessuno dei giocatori che possiede il pulsante di prenotazione del Velox (di colore giallo, numerato da uno a sei), cioè nessuna delle persone in gara, deve premere per prenotarsi: se ciò si verifica, l'annullamento automatico della prova con la comparsa del messaggio "Non si può premere prima del via!", nel caso in cui più di un giocatore abbia premuto anticipatamente, oppure dell'altro messaggio, "Il n. 2 ha premuto prima del via!" se uno soltanto dei partecipanti (nella fattispecie quello del pulsante numero due) abbia sbagliato prenotandosi prima del tempo. Se tutto procede regolarmente allo scadere del tempo prefissato appare sul display il messaggio, "Premere per prenotarsi!", che abilita i pulsanti di prenotazione al rego-

lare funzionamento: il primo dei concorrenti che preme il proprio pulsante ha il diritto alla risposta, se si tratta di un quiz-game, oppure è stato il più veloce se si effettua una competizione per la verifica della prontezza di riflessi.

Alla fine di ogni singola prova, valida o no che sia, basta premere il tasto rosso Reset collegato all'apparecchio Velox per riprendere: un nuovo e diverso tempo di attesa viene subito selezionato. Ognuno dei pulsanti di risposta (ce ne possono essere fino a sei oltre al Reset) è collegato al Velox via cavo, in modo da permettere a ogni persona di giocare a giusta distanza dal computer e dal video, e senza essere disturbata e ostacolata dai rimanenti giocatori.

Durante il funzionamento del programma anche il dispositivo Velox svolge un'appropriata azione di monitoraggio assai utile all'utente tramite i due led di colore giallo Free-line e rosso Reset che, accendendosi, segnalano all'istante rispettivamente la condizione di linea libera e l'azionamento del sistema di Reset.

Daniele Malvasi



MasterCard

Con questo software potrete far funzionare al meglio l'apparecchio elettronico con lettore ottico, per la gestione di tutte le operazioni relative a tessere a strip codificato, pubblicato sul numero di luglio/agosto.

Questo eccezionale programma contiene al suo interno complesse routine di elaborazione, e permette di ottimizzare le prestazioni del sistema hardware-software composto dal computer Commodore 64 e dal dispositivo elettronico MasterCard (applicato al computer attraverso la porta utente), riuscendo a gestire in modo professionale e totalmente automatico tutte le operazioni relative alla lettura di tessere (formato universale card) codificate con il metodo a barre seriali nere e bianche, rilevando in pochi decimi di secondo i punti di inizio e fine scanning, oltre al verso di inserimento nel lettore ottico a doppia fotoresistenza. È pertanto possibile, tramite un semplice inserimento di una card precodificata, rilevarne non solo il numero corrispondente di identificazione ma anche lo strip seriale a 15 barre (4 di chiave e 11 di personalizzazione). Routine supplementari permettono poi di effettuare, oltre alle operazioni di creazione, codifica e decodifica delle card, anche memorizzazioni per nominativi di riferimento e movimentazioni di in-

put/output (da 0 a 999.999 unità per ogni codice), sfruttando un capace archivio predisposto per ben 100 tessere. In pratica il software è in grado di elaborare tutti i segnali impulsivi che determinano, attraverso i due input PB0 e PB1 della porta utente del computer, la corretta sequenza seriale di lettura per ogni singola tessera sottoposta a scanning. Tramite un'unica videata grafica multicolor si può controllare con facilità lo svolgimento di tutte le routine di elaborazione, con monitoraggio effettuato da un multidisplay che permette una lettura istantanea e soprattutto assicura facilità di interpretazione; il tutto senza rischio di errori o di eccessive approssimazioni.

Funzionamento e uso

Terminata l'operazione di caricamento, non appena il programma parte (loading con autostart) o viene fatto partire (tramite un Run L di lancio) compare la videata-monitor a fondo e bordo di colore nero, che rimane invariata nella sua parte grafica e struttu-

rale durante tutte le fasi di elaborazione delle routine. Nella parte superiore dello schermo sono presenti il nome MasterCard 2 in colore giallo e il data-set di copyright in arancio.

L'area-video rimanente viene suddivisa in tre distinti settori: il primo (azzurro) comprende due riquadri con il doppio display per la visualizzazione dei codici a barre e di quelli numerici corrispondenti; un secondo settore (verde e centrale) è il video di proiezione della parte di archivio (memoria tessere) in scanning; il terzo (in grigio e nella parte inferiore dello schermo) contiene i comandi di controllo principali (tasti-funzione) per le varie operazioni di possibile effettuazione.

Il doppio display del primo settore visualizza sulla parte sinistra le corrispondenze dei codici tessera, riportando con precisione lo strip seriale dei 15 segni bianchi o neri (rispettivamente neri e azzurri sul video):

CORRISPONDENZA
SCAN #####
DATA

Sulla parte destra è invece visibile il numero a quattro cifre del codice tessera di riferimento, variabile da 0000 a un massimo di 2047 (da 0001 a 0100 è invece compreso il range numerico di gestione):

CODICE CARD

....

Tra sequenza a barre e numero di codice visualizzati esiste sempre piena corrispondenza, sia in operazioni di codifica e decodifica sia di scanning (monitoraggio esplorativo) su archivio. La zona inferiore del video comprende il menù con le quattro principali opzioni che permettono all'utente di creare nuove tessere, di codificare e decodificare (le conversioni sono automatiche) e di effettuare operazioni di variazione su un archivio capace di contenere fino a 100 distinti nominativi e altrettanti contatori da 0 a 999999 unità:

- F1 - Lettura ottica
- F3 - Codifica
- F5 - Memorizzazione
- F7 - Carica.

Premendo F1 si abilita la routine che permette di leggere i dati della card tramite rilevazione ottica. A tal scopo ogni tessera va inserita nel lettore indifferentemente da destra o da sinistra: il senso di scorrimento non ha importanza perché una speciale procedura di conversione automatica provvede, analizzando i codici chiave, a formattare il file seriale di input. Ovviamente il discorso non vale per il fronte/retro: infatti gli strip a barre bianche e nere, ossia la facciata della tessera, devono essere rivolti verso il lettore ottico e cioè verso i led dell'apparecchio Mastercard, inoltre non capovolti (la parte col codice deve in pratica entrare all'interno del binario di scorrimento). Ogni scanning di card va eseguito attenendosi scrupolosamente all'osservanza di due fondamentali principi: scorrimento costante

della card a velocità moderata (non più di 1 cm/sec) e disponibilità a ripetere anche più volte l'operazione se il computer evidenzia un errore di lettura. Il problema dell'affidabilità, per quanto teoricamente risolto in sede progettuale, per dispositivi di questo tipo non arriva mai al 100% di operazioni riuscite al primo colpo (questo accade anche in lettori ottici molto complessi e costosi): una conservazione premurosa delle tessere e l'acquisizione di un buon modus operandi permettono comunque all'utente anche inesperto di raggiungere, almeno con Mastercard, risultati di prim'ordine. Per ogni lettura ben eseguita viene subito proiettato, nell'apposito riquadro azzurro di destra, il codice a quattro cifre corrispondente alla card sottoposta a scanning ottico: può variare da 0000 (11 caselle dello strip a barre tutte bianche) a 2047 (caselle tutte annerite), per un totale di ben 2048 tessere gestibili (e di queste ben 100 anche integralmente su archivio, con codice da 0001 a 0100). La comparsa di quattro asterischi (****) indica invece la necessità di ripetere l'operazione di lettura o di correggere lo strip della card (perché macchiato o usurato): ovviamente anche la comparsa di un codice strano o diverso deve indurre l'utente a verificare l'attendibilità della lettura e lo stato di conservazione della tessera. Durante lo scanning una subroutine sonora provvede a segnalare anche acusticamente l'input dei dati in lettura: un bip più acuto evidenzia la fine dell'operazione. È possibile rilevare in tempo reale lo strip seriale corrispondente ai dati in lettura, che viene proiettato in sequenza sul display azzurro di sinistra: caselle piene e azzurre corrispondono ai neri del codice a barre, caselle vuote e trasparenti ai bianchi. Premendo F3 si seleziona invece la procedura di creazione e codifica delle tessere. Tutto avviene in base a un procedimento assai semplice: inserendo il numero di attribuzione tramite i 10 tasti numerici dall'1 allo

0. Valori superiori a 2047 vengono rifiutati, mentre numeri non di quattro cifre non vengono accettati (ad esempio per codificare col numero 46 si digita 0046). Il codice è visibile nel riquadro Codice Card: se regolarmente inserito causa l'immediato calcolo dello strip a barre corrispondente (sezione Data), che può essere eventualmente ricopiato sulla card da personalizzare. La codifica comprende sempre 15 caselle, di cui le prime e le ultime due sono fisse (di chiave) e le altre 11 centrali variabili: il codice minore, 0000, genera 11 caselle bianche (vuote), mentre a quello massimo, 2047, corrispondono 11 caselle nere (piene, azzurre sul video). Un infallibile metodo procedurale è quello di verificare subito la corretta scrittura delle tessere facendole leggere otticamente e controllando che il codice fornito sia lo stesso di quello attribuito. La scrittura delle card va effettuata con un buon pennarello nero a punta fine, rispettando rigorosamente la delimitazione delle singole caselle. È ovviamente meglio evitare doppi (tessere con codici uguali): d'altra parte è possibile creare oltre 2.000 pezzi (di cui 100 archiviabili), più che sufficienti per computerclub, circoli, associazioni e gestioni di piccoli e medi stock (magazzini, campionari, gruppi di persone, ecc.). L'archivio in dotazione contiene ben 100 campi e permette di gestire integralmente altrettante tessere che siano codificate nel range previsto da 0001 a 0100; l'archivio è raggiungibile tramite chiamata diretta di un certo codice (opzione Codifica con F3), oppure automaticamente inserendo una card (opzione Lettura ottica con F1). In particolare in base all'indirizzo di riferimento fornito la visualizzazione della zona di archivio interessata viene effettuata sul settore centrale dello schermo, di colore verde, che è in pratica un display a tre linee (queste in colore celeste). Inizialmente l'indirizzo assunto è quello iniziale, ovvero il codice 0001 (visualizzato al centro come 001):

001 ===== 000000
002 ===== 000000

Il display contiene per ogni linea il numero di codice, il nominativo di identificazione (spazio disponibile 23 caratteri) e una quantità di riferimento (da 000000 aumentabile fino al massimo di 999999). Come indicano i due marker di colore viola, la linea di riferimento è sempre quella di mezzo, mentre le altre due sono quella precedente (la superiore) e quella seguente (l'inferiore) nell'ambito dell'archivio di riferimento: infatti indirizzando per esempio il numero 0038, si otterrà la visualizzazione delle linee di archivio 037, 038 e 039. Con codici superiori o inferiori al range previsto (minimo 0001 e massimo 0100) verrà abilitata la zona di estremo inferiore (***) - 001 - 002) o superiore (099 - 100 - ***) dell'archivio stesso. Oltre all'indirizzamento diretto è possibile effettuare comodamente lo scanning passo-passo dell'archivio sia retrocedendo sia avanzando, premendo rispettivamente i tasti abilitati < (variazione -1) oppure >

(variazione +1). Spostamenti più rapidi sono inoltre possibili con Shift + < (variazione -10) e Shift + > (variazione +10): anche in questo caso, come sempre, come linea di riferimento vale quella centrale delle tre visualizzate nell'apposito display. Nell'ambito dell'archivio per ognuno dei 100 numeri a disposizione è definibile un nominativo di corrispondenza e identificazione che può essere al massimo di 23 caratteri: sono abilitati a tal scopo i tasti alfabetici (da A a Z), quelli numerici (da 0 a 9), lo spazio (), il trattino (-) e il segno di uguaglianza (=).

L'input avviene in modo diretto sulla linea di riferimento del display, senza bisogno di conferma, nella posizione segnata dai due cursori di colore viola che si spostano sulla destra a ogni immissione valida: per correggere eventuali errori, o comunque per effettuare modifiche a quanto già memorizzato, basta resettare i cursori stessi riportandoli sulla sinistra di linea tramite il tasto con la freccetta a sinistra (posto a fianco di quello numerico 1 sulla prima fila della tastiera del Commodore). Il Reset dei cursori avviene comunque sempre a ogni variazione di proiezione del display, anche durante lo scanning. I nominativi da attribuire ai codici possono essere di molti tipi: per esempio saranno cognomi e nomi di iscritti se si gestisce la

lista di un club, oppure denominazioni commerciali relativamente a prodotti catalogati come magazzino, o addirittura altri codici di corrispondenza per password, comunicazioni segrete, e quanto altro fantasia e necessità suggeriscono all'utente. Per ognuno dei 100 nominativi è inoltre disponibile un contatore capace di

999.999 unità, utilizzabile a piacere, per esempio per memorizzare la quantità disponibile a magazzino di un certo articolo, oppure la spesa per una determinata voce del bilancio familiare, o l'età di una corrispondente persona. Le variazioni quantitative avvengono sulla linea di riferimento proiettata, e possono essere fatte in aumento o in diminuzione rispettivamente coi tasti] (variazione +1) e [(variazione -1).

Anche in questo caso l'apporto del tasto Shift, in abbinamento ai due già menzionati, permette maggiori escursioni: +10 con Shift +] e -10 con Shift + [. Quantità minori o maggiori del range previsto non saranno prese in considerazione dai contatori. È possibile memorizzare permanentemente su cassetta tutti gli archivi di volta in volta creati, modificati o aggiornati, predisponendo il registratore su Rec e premendo semplicemente il tasto-funzione F5: lo schermo diventa completamente nero per 133 secondi circa (tempo necessario al trasferimento dei dati), dopodiché avviene il ritorno automatico alla videata di riferimento. È ovviamente consigliabile l'uso di audiocassette di breve durata e soprattutto dedicate, contenenti cioè solo l'archivio salvato.

Il programma vero e proprio va invece conservato su un'altra cassetta.

Il tasto funzione F7 permette l'operazione opposta al salvataggio, e cioè la carica su computer, dal registratore in Play di un file (archivio) già creato, caratterizzato e salvato: anche con questa opzione il tempo occorrente è di circa 113 secondi.

Durante il funzionamento del programma anche il dispositivo Mastercard svolge un'appropriata azione di monitoraggio assai utile all'utente tramite i due led di colore rosso Scan e Data che accendendosi segnalano all'istante, rispettivamente, lo scorrere della tessera nel lettore ottico e l'input seriale dei dati di decodifica.

Daniele Malavasi



Pagina mancante (pubblicità)

Hardcopy dello schermo hires

Vorrei porvi un problema che ho incontrato fin da quando ho iniziato a sviluppare programmi che utilizzano l'alta risoluzione.

Fino a quando si utilizzano potentissime espansioni del Basic, come il Simons' Basic, anche la hardcopy dello schermo può essere risolta con un semplice comando, ma quando per i motivi più svariati non si può ricorrere a questi tools, il problema della copia su carta dello schermo hires è veramente serio, anche perché bisogna conciliare le caratteristiche dei vari computer Commodore con i vari modelli di stampanti sul mercato. Vorreste colmare questa lacuna e fornire un software adeguato allo scopo?

Francesco Gentilini
Brescia

Una delle più potenti caratteristiche dei computer Commodore è la grafica. Naturalmente quando

il computer viene spento ogni immagine viene persa. La soluzione è quella di trasferire su carta il contenuto dello schermo per mezzo della stampante, in poche parole l'hardcopy dello schermo hires. La routine che proponiamo a questo scopo e di cui pubblichiamo i listato è la più potente e versatile tra quelle mai presentate su questa rivista poiché funziona con ogni stampante compatibile con la Commodore 1525 e con ogni tipo di computer Commodore: 128, 64, Plus/4 e C16. Hires dump è nella forma caricatore Basic. Se avete intenzione di utilizzare questo programma sul 128, assicuratevi di utilizzare il modo 128 quando digitate il programma, in modo tale da riuscire a tokenizzare le keyword specifiche del 128. Dopo aver digitato il programma dovete informare la routine circa la posizione della mappa hires nella memoria. Per fare questo dovete aggiornare la variabile SL alla linea 30. Per il 128, Plus/4 e il C16 u-

tilizzate il valore 8192. Questo è l'indirizzo dello schermo di alta risoluzione quando utilizzate i comandi grafici del Basic. Se invece utilizzate un 64 questo indirizzo dipenderà dal particolare programma con cui lo avete disegnato o dalle caratteristiche del programma che state scrivendo, in ogni caso dovete inserire l'indirizzo iniziale. Molti programmi di disegno per Commodore 64 utilizzano l'area che inizia da 24576 (\$6000). Se avete problemi a localizzare l'area, potrebbe aiutarvi ricordare che l'indirizzo iniziale deve essere un multiplo pari di 8192 (\$2000). A questo punto potete salvare il programma su disco. Quando siete pronti per stampare il vostro disegno con il 64 o il 128, caricate il programma e date il Run. Se state usando un Plus/4 o un C16, digitate POKE56,23:POKE5,0:CLR e quindi caricate e fate girare il programma. Vi verrà chiesto dove volete memorizzare la routine in linguaggio macchina. La risposta che dovete fornire dipende dal calcolatore che state usando, per esempio:

Commodore 64: 49152
Commodore 128: 3072
Plus/4 o C16: 5888

Quando la routine in linguaggio macchina è stata posta in memoria, digitate New e caricate la schermata di alta risoluzione che volete stampare, per esempio con un comando del tipo LOAD"SCHERMO",8,1 oppure se utilizzate il 128 con BLOAD"SCHERMO",B0,P8192, naturalmente dovete rimpiazzare la parola SCHERMO con il nome con cui avete salvato su disco la vostra schermata. Per inviare il disegno in stampa è sufficiente dare la SYS corrispondente all'indirizzo iniziale di allocazione della routine in linguaggio macchina (vedi elenco precedente). Per esempio per Commodore 64 l'istruzione è SYS49152. Al termine della stampa, la stampa resta settata in modo grafico, spegnetela e riaccendetela per resettarla.

Listato 1.

```
10 poke53280,0:poke53281,0
30 sl=8192:rem locazione dello schermo hires
40 printchr$(147)chr$(5)" copyright 1988 gruppo editoriale jce"
50 print:print
70 input"indirizzo iniziale":s
80 fori=stos+237:readd:pokei,d:x=x+d:next
90 ifx<>34823thenprint:print:print"errore nei data"
100 sl=s+7671:hb=int(sl/256):lb=sl-256*hb
110 pokes+57,lb:pokes+61,hb:pokes+65,lb+8:pokes+69,hb
120 print:print:print"hires dump installato":print"sys"s"per attivare"
130 data 169,0,32,189,255,169,1,162
140 data 4,160,0,32,186,255,32,192
150 data 255,169,0,32,189,255,169,21
160 data 162,4,160,21,32,186,255,32
170 data 192,255,162,1,32,201,255,169
180 data 8,32,210,255,169,0,133,80
190 data 133,82,133,83,169,25,133,81
200 data 169,247,133,250,169,61,133,251
210 data 169,255,133,252,169,61,133,253
220 data 160,8,177,250,133,254,177,252
230 data 133,255,166,80,240,7,70,254
240 data 102,255,202,208,249,162,8,102
250 data 255,38,254,202,208,249,165,83
260 data 240,4,37,254,133,254,169,128
270 data 5,254,32,210,255,136,208,210
280 data 240,3,24,144,203,56,165,250
290 data 233,64,133,250,165,251,233,1
300 data 133,251,56,165,252,233,64,133
310 data 252,165,253,233,1,133,253,198
320 data 81,208,173,169,25,133,81,169
330 data 13,32,210,255,230,80,165,80
340 data 201,8,208,14,169,0,133,80
350 data 162,8,198,250,198,252,202,208
360 data 249,24,165,250,105,72,133,250
370 data 165,251,105,31,133,251,24,165
380 data 252,105,72,133,252,165,253,105
390 data 31,133,253,230,82,165,82,201
400 data 45,144,159,201,46,240,6,169
410 data 31,133,83,208,149,32,204,255
420 data 169,1,32,195,255,96

ready.
```


Il mensile con disco programmi per C64 e C128

Gruppo Editoriale
JCE

COMMO DISK

GIOCO

Mille nemici
nel mondo
in miniatura

UTILITY

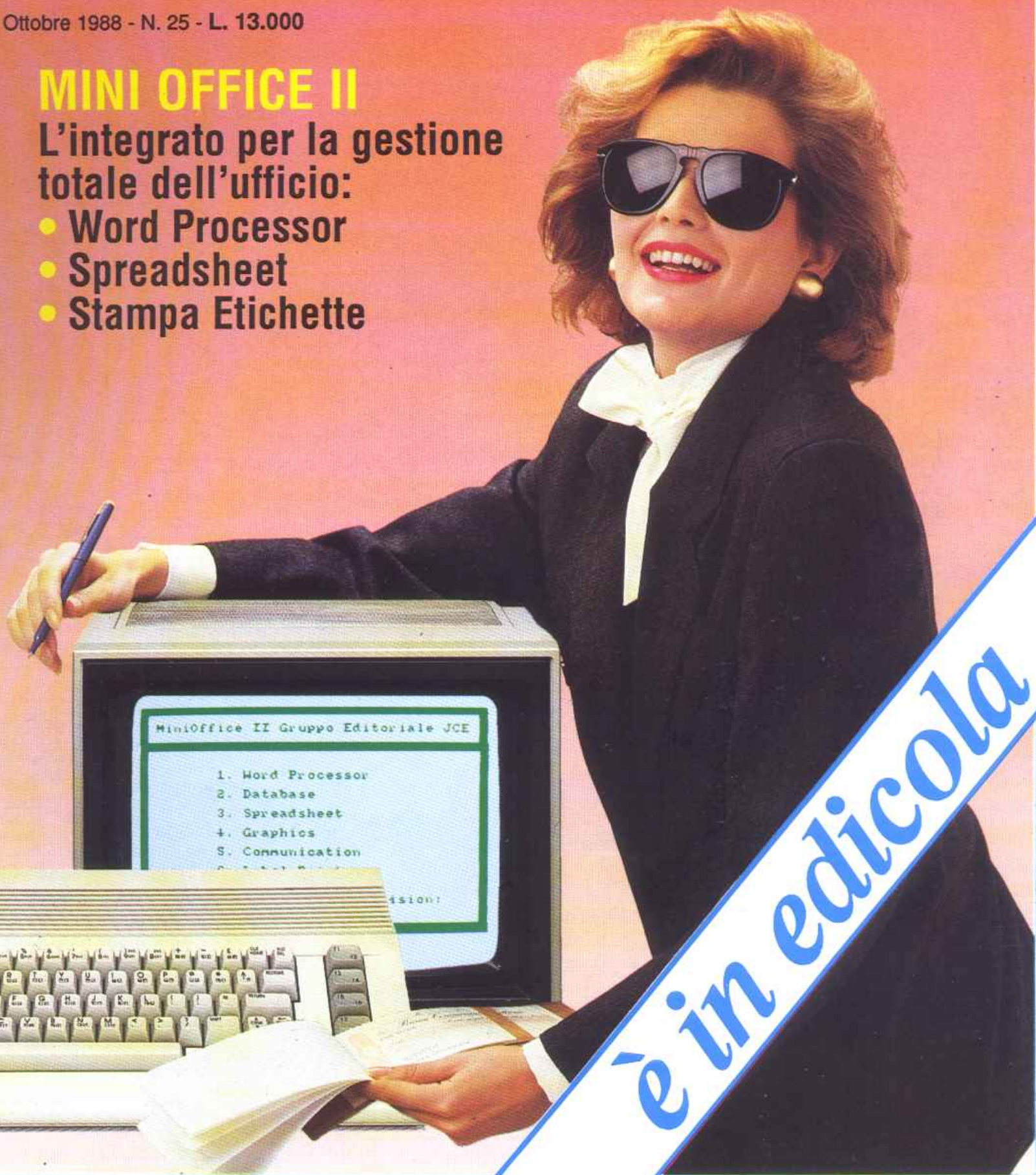
Con Smart Editor tutte
le funzioni del C128

Anno II - Ottobre 1988 - N. 25 - L. 13.000

MINI OFFICE II

L'integrato per la gestione
totale dell'ufficio:

- Word Processor
- Spreadsheet
- Stampa Etichette



è in edicola

Pagina mancante (pubblicità)